

Figure 1

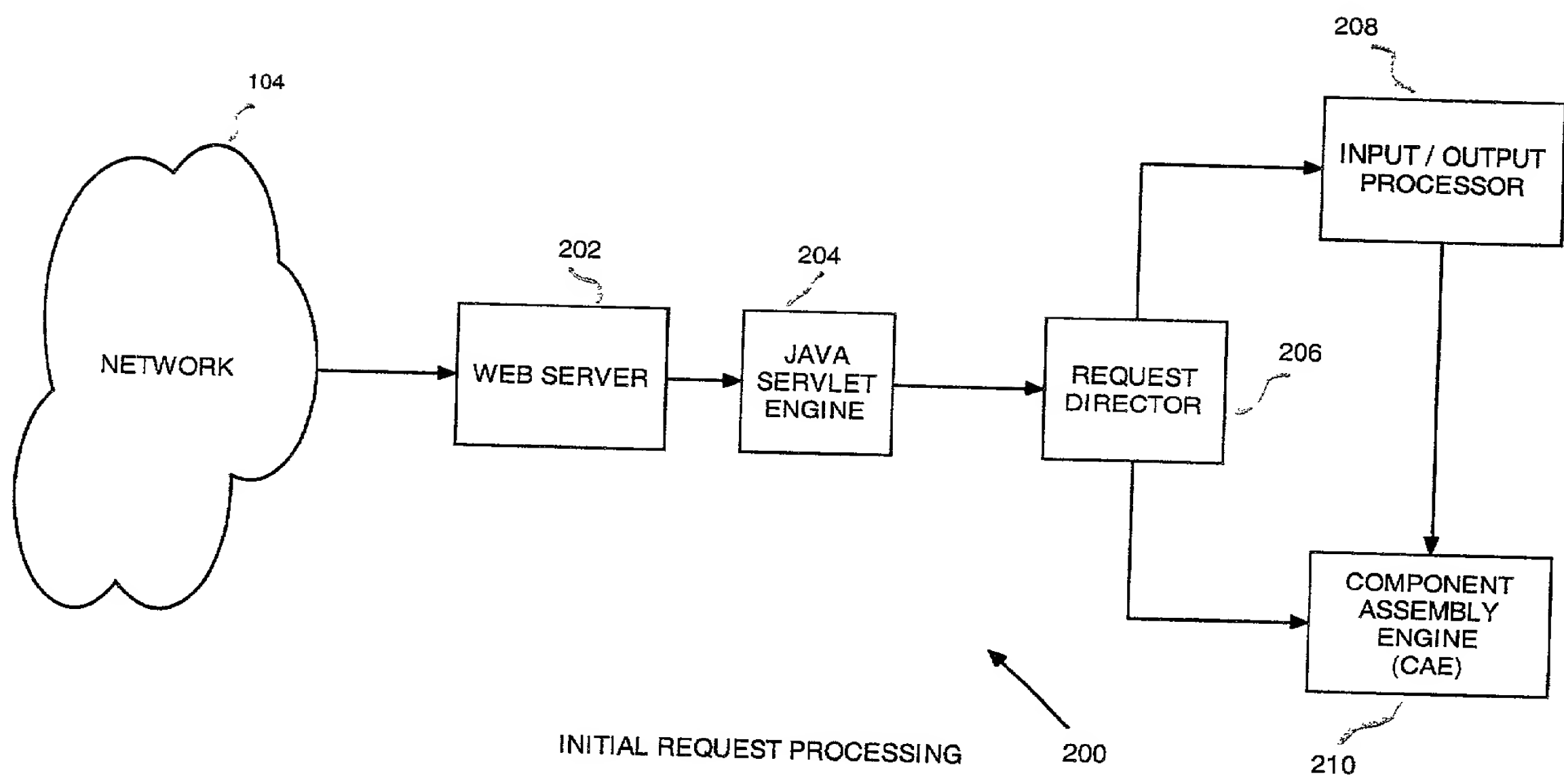


Figure 2

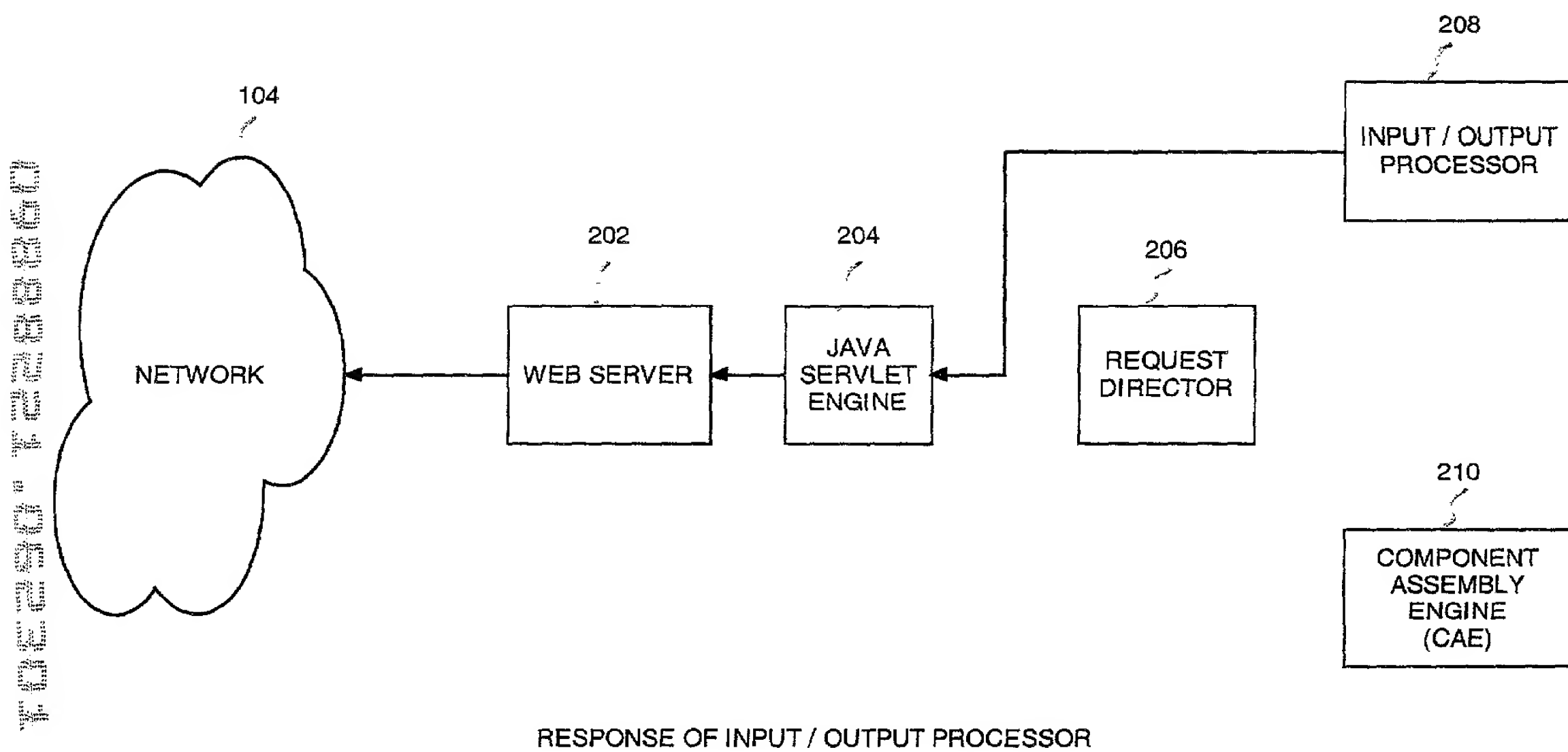
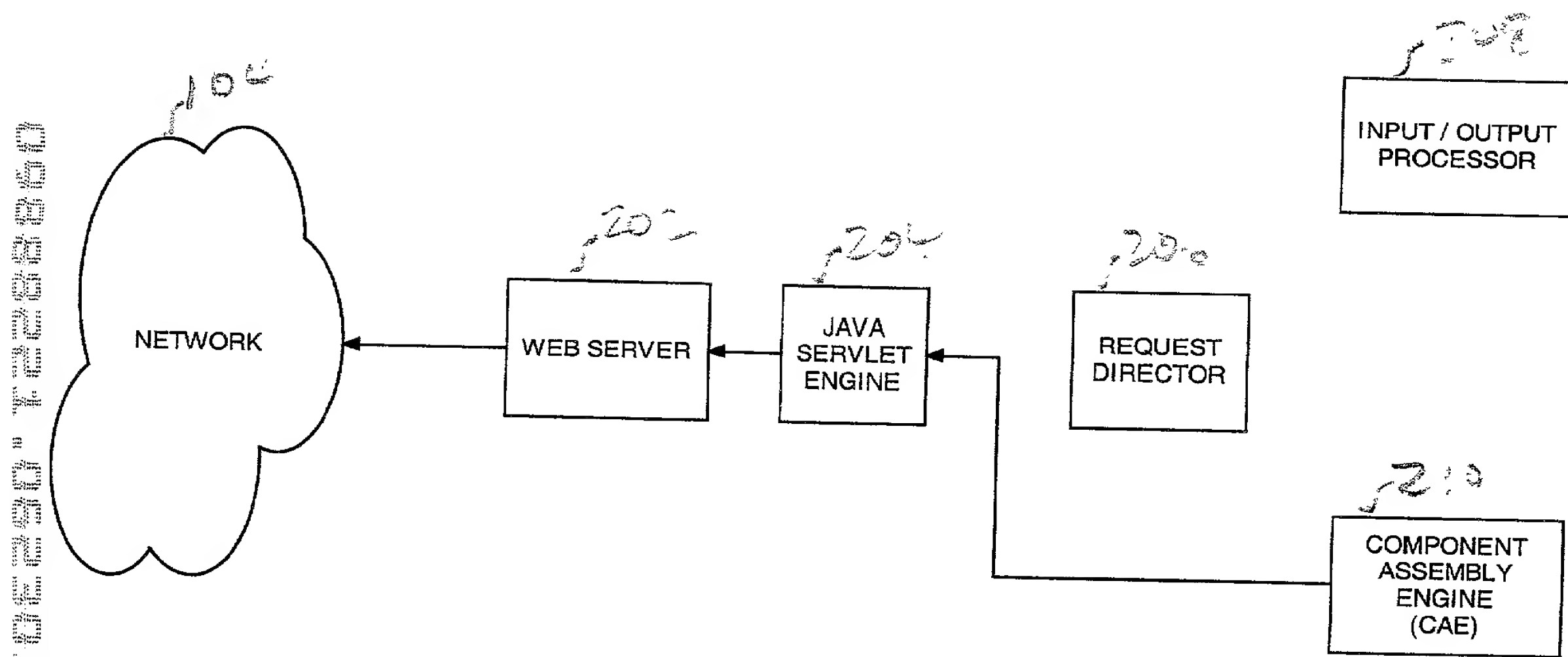
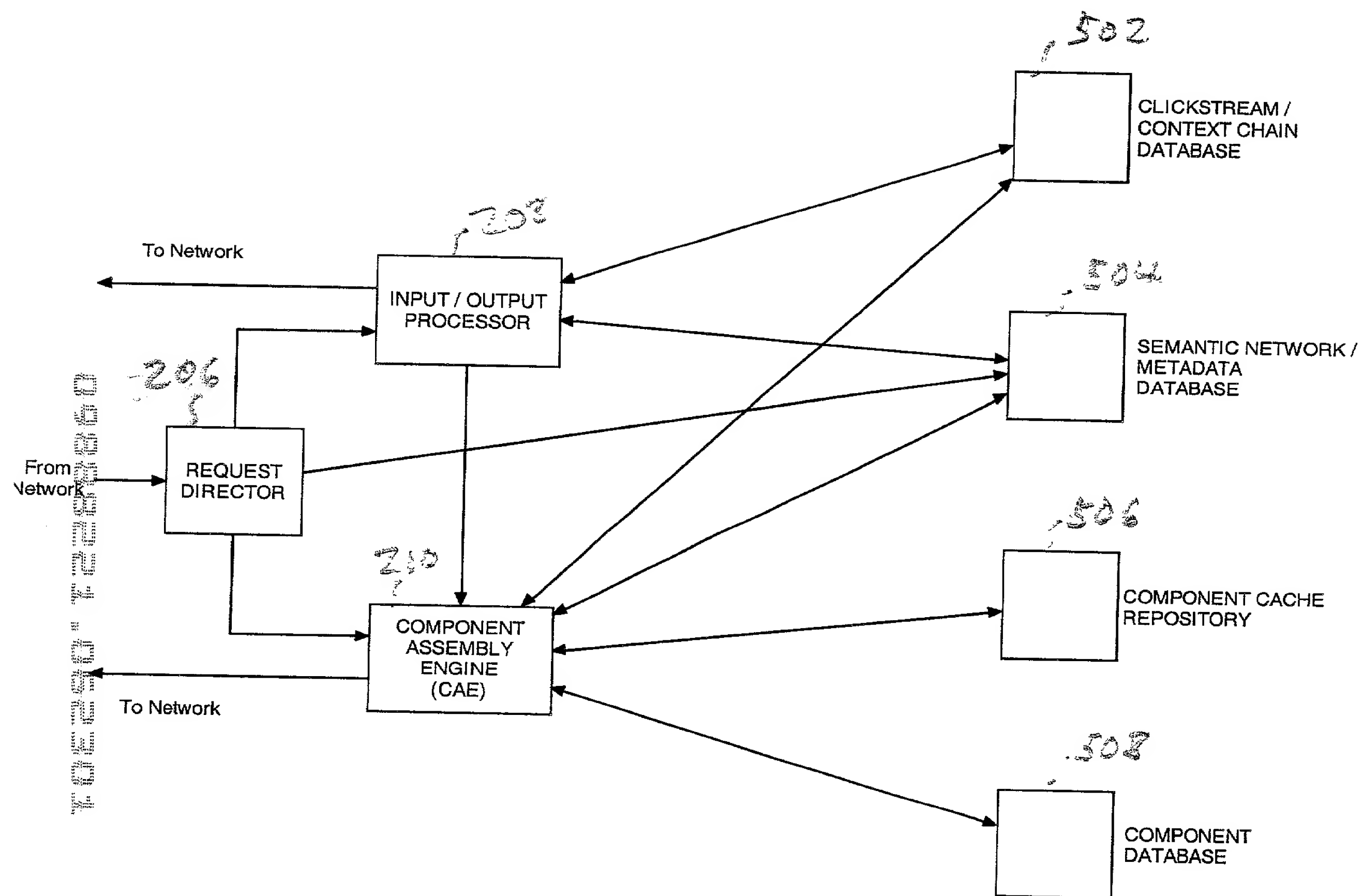


Figure 3



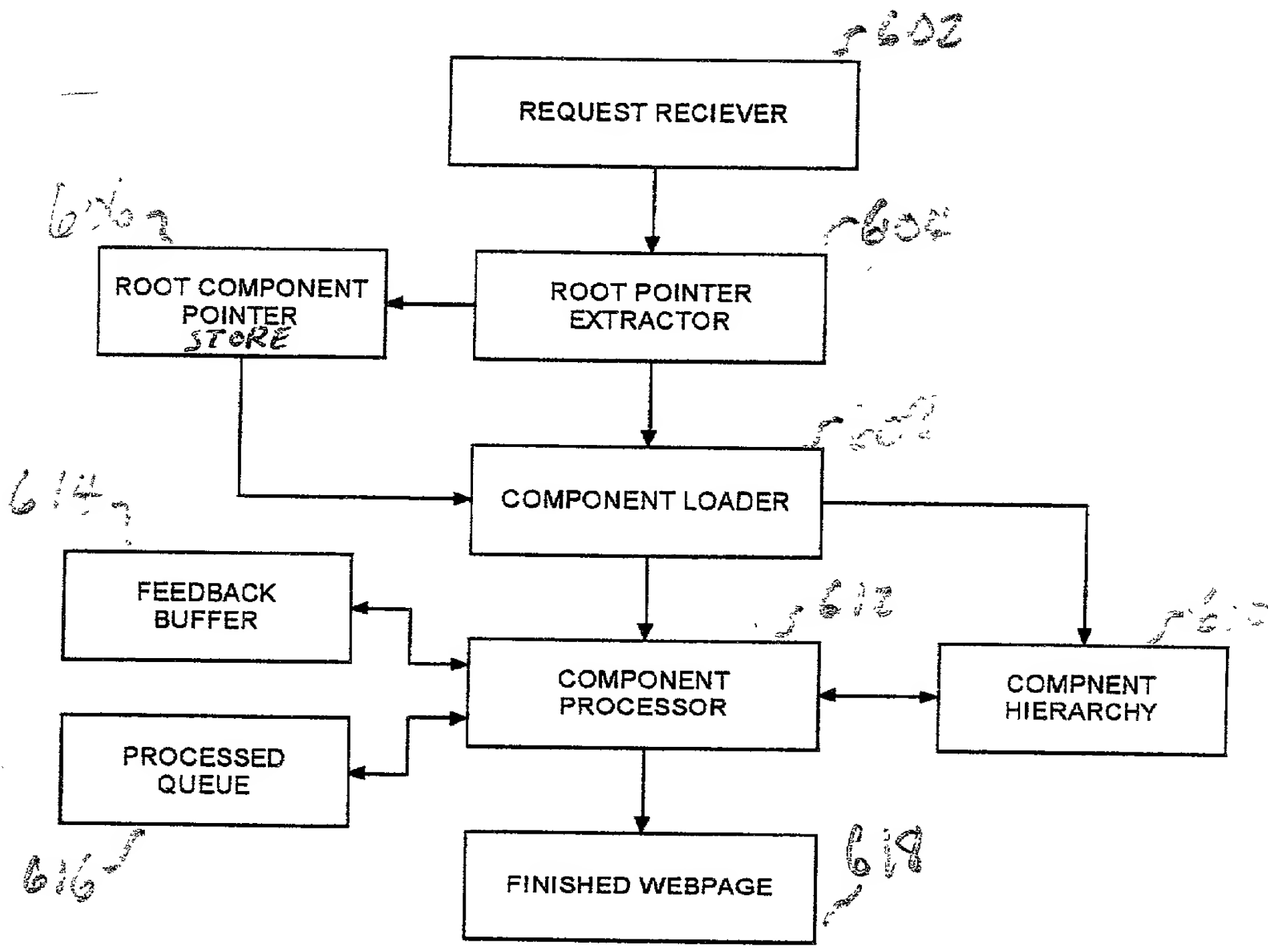
WEBPAGE DELIVERY

Figure 4



BACKEND DATABASES

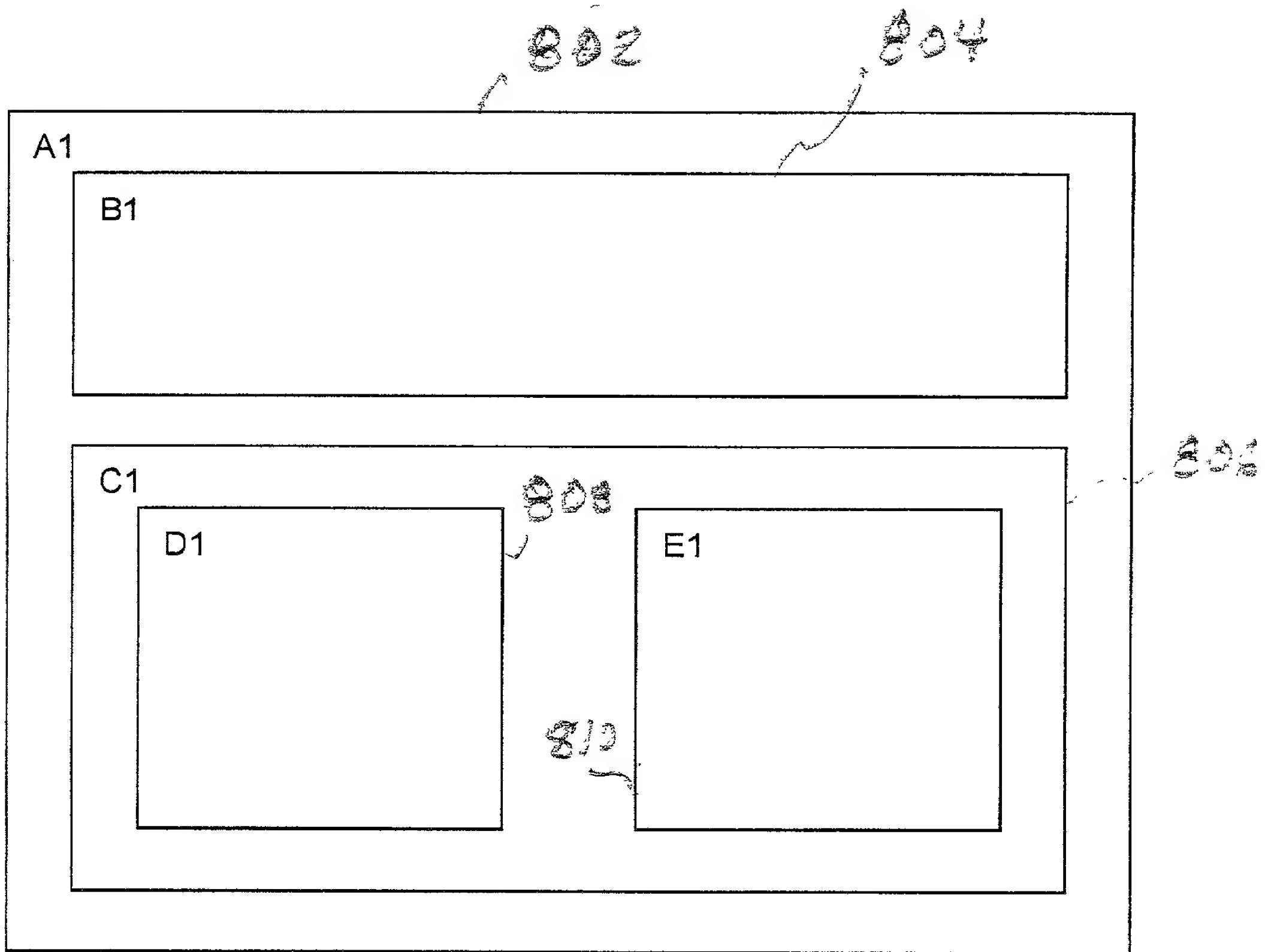
Figure 5



COMPONENT ASSEMBLY ENGINE (CAE)

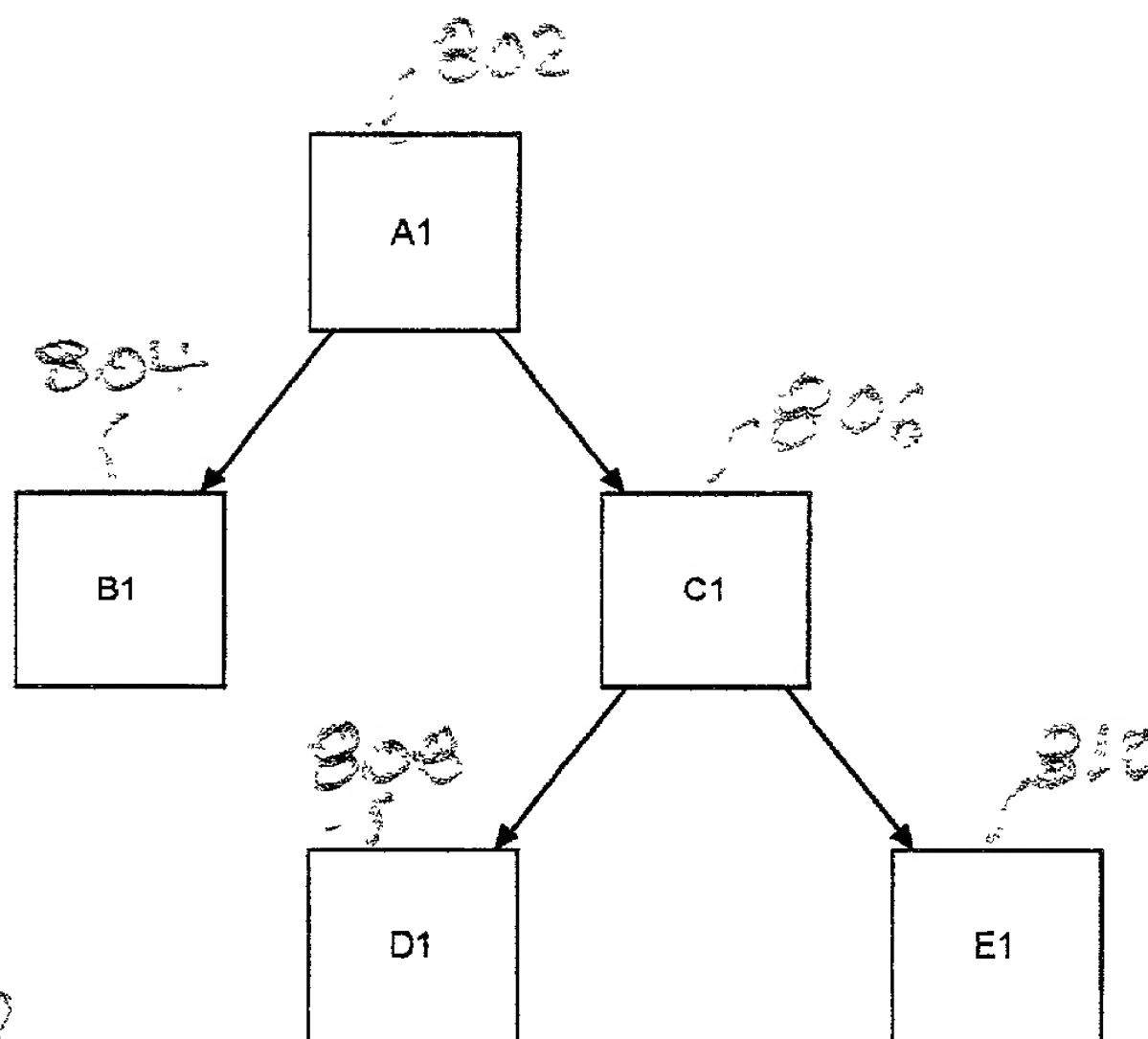
Figure 6

4.06.2020 12:28:50



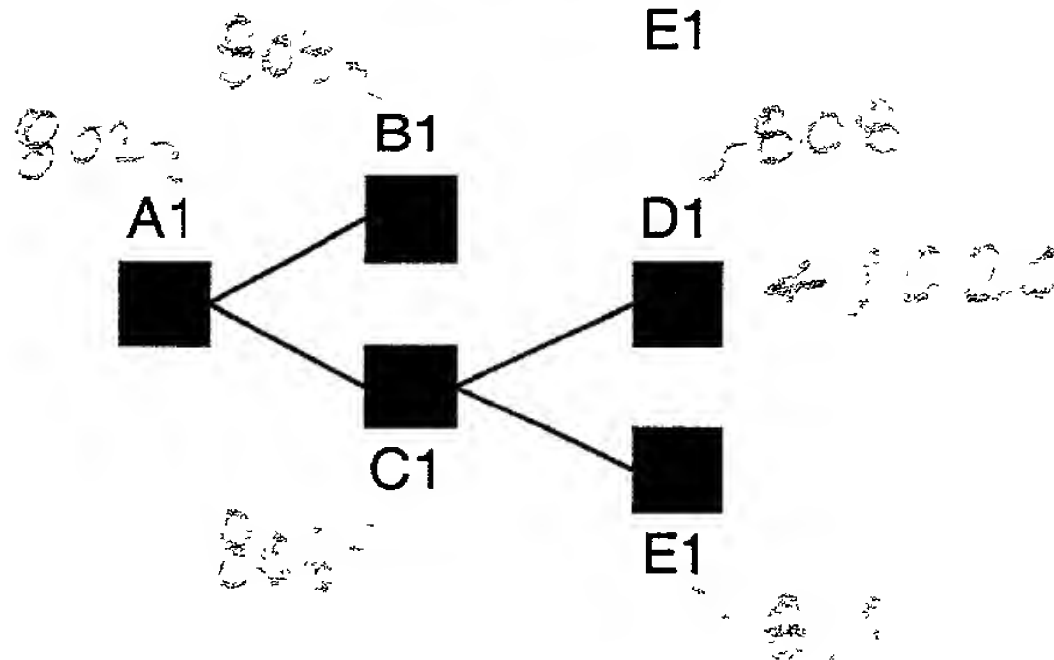
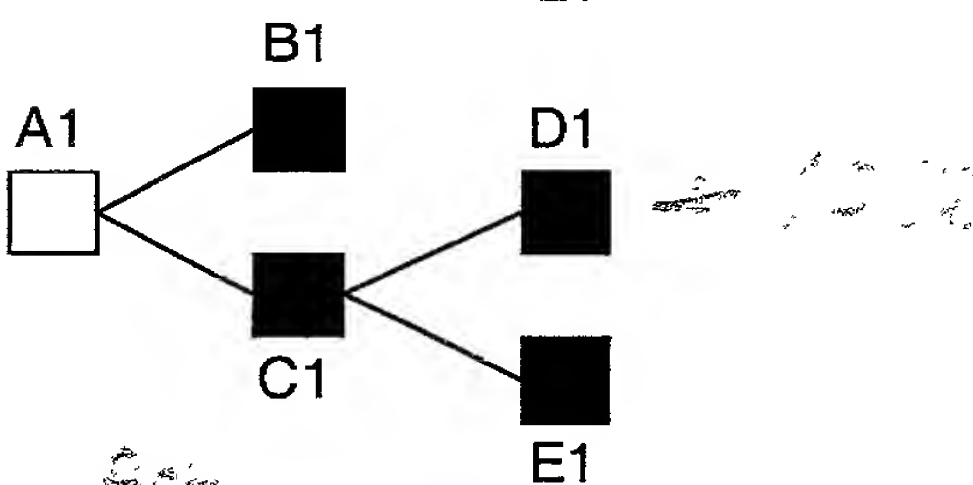
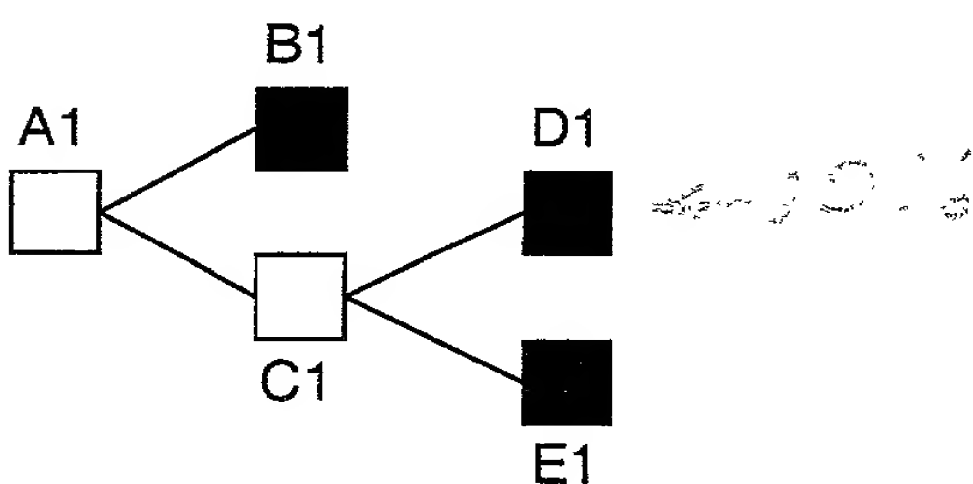
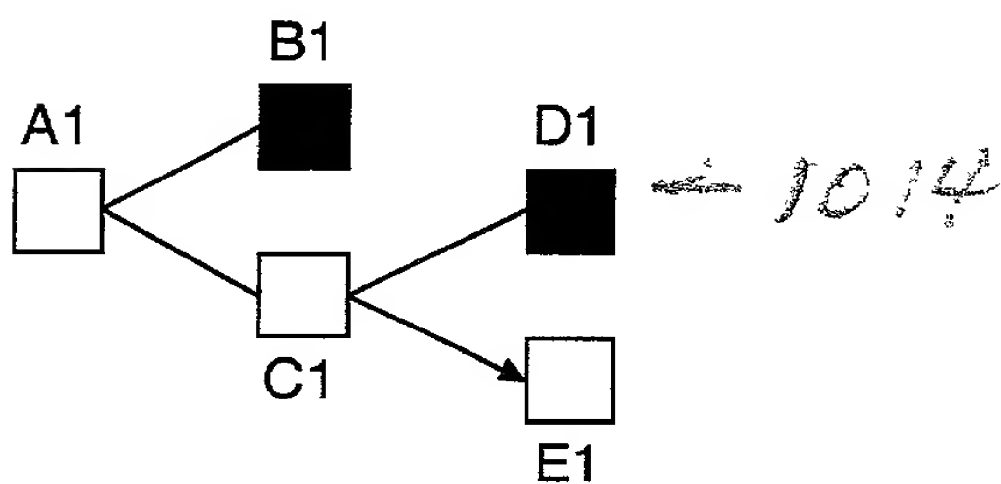
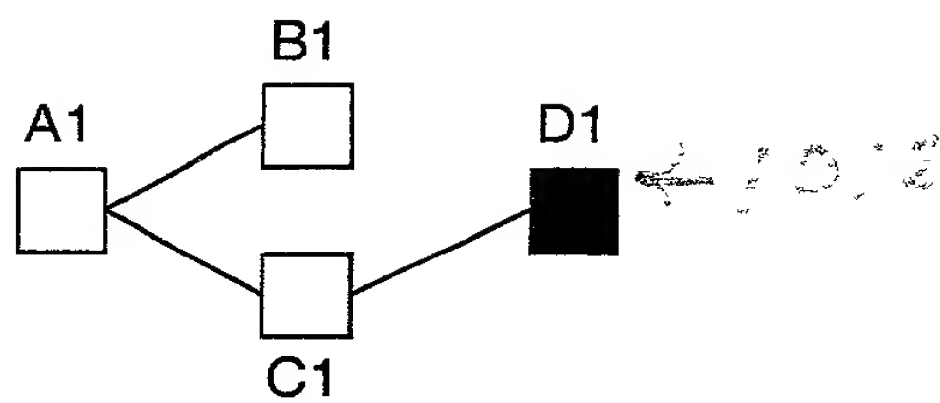
COMPONENT MAKEUP OF EXAMPLE WEBPAGE

Figure 8



EXEMPLARY HIERARCHICAL TREE REPRESENTATION

Figure 9

[illegible]

COMPONENT LOADING AND PROCESSING

Figure 10

The diagram illustrates the decomposition of a 2D component into replacement and supplementation components. The top part shows a tree structure with nodes A2, B2, C2, and D2 (shadow component). Two arrows point down to 'REPLACEMENT' and 'SUPPLEMENTATION' sections. 'REPLACEMENT' shows a tree structure with nodes A2, B2, C2, E2, and F2. 'SUPPLEMENTATION' shows a tree structure with nodes A2, B2, C2, D2, G2, and H2. Handwritten binary codes (1105, 1110, 1112, 1114, 1116, 1118, 1120, 1122) are written next to the nodes. A handwritten '7' is at the bottom left, and '1105' is at the bottom right.

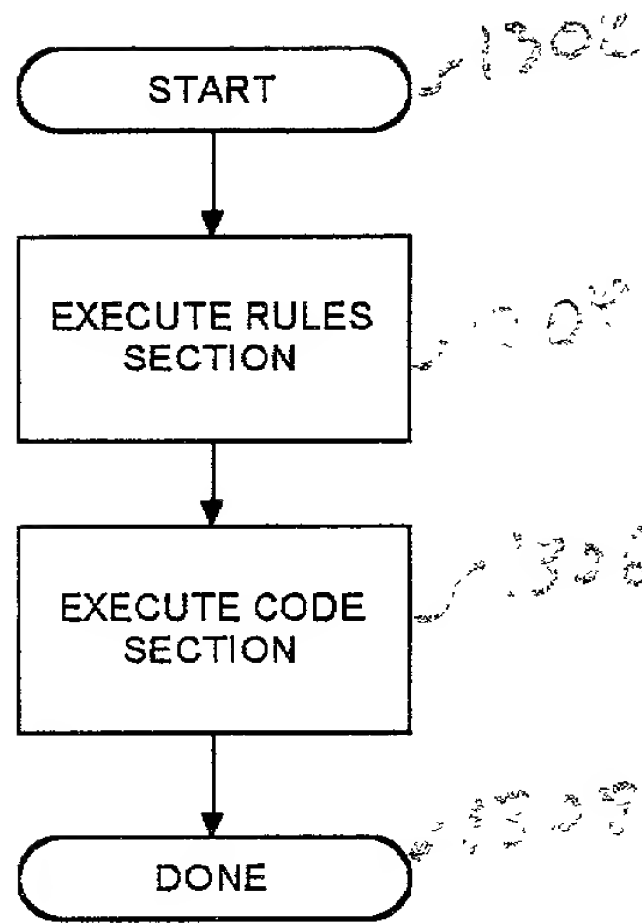
Figure 11

```

graph TD
    A3_1203[A3 1203] --> B3_1210[B3 1210]
    A3_1203 --> C3_1212[C3 1212]
    C3_1212 --> D3_1211[D3 1211 shadow component]
    D3_1211 --> A3_1208[A3 1208]
    D3_1211 --> A3_1206[A3 1206]
    A3_1208 --> B3_1210_2[B3 1210]
    A3_1208 --> C3_1212_2[C3 1212]
    C3_1212_2 --> E3_1211[E3 1211]
    C3_1212_2 --> F3_1218[F3 1218]
    A3_1206 --> B3_1210_3[B3 1210]
    A3_1206 --> C3_1217[C3 1217]
    C3_1217 --> G3_1220[G3 1220]
    C3_1217 --> H3_1232[H3 1232]
  
```

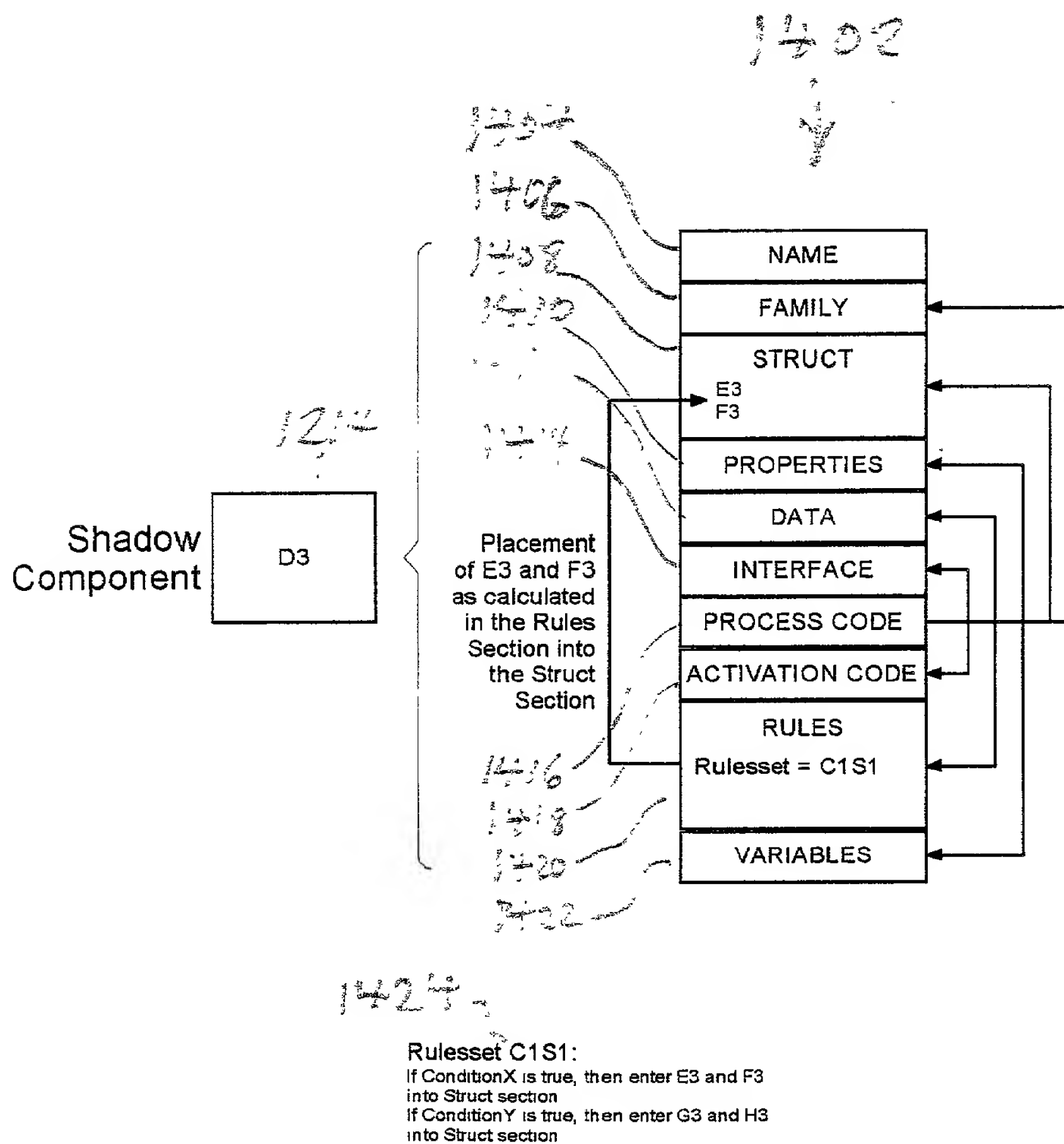
SHADOW COMPONENT RESOLUTION

Figure 12.



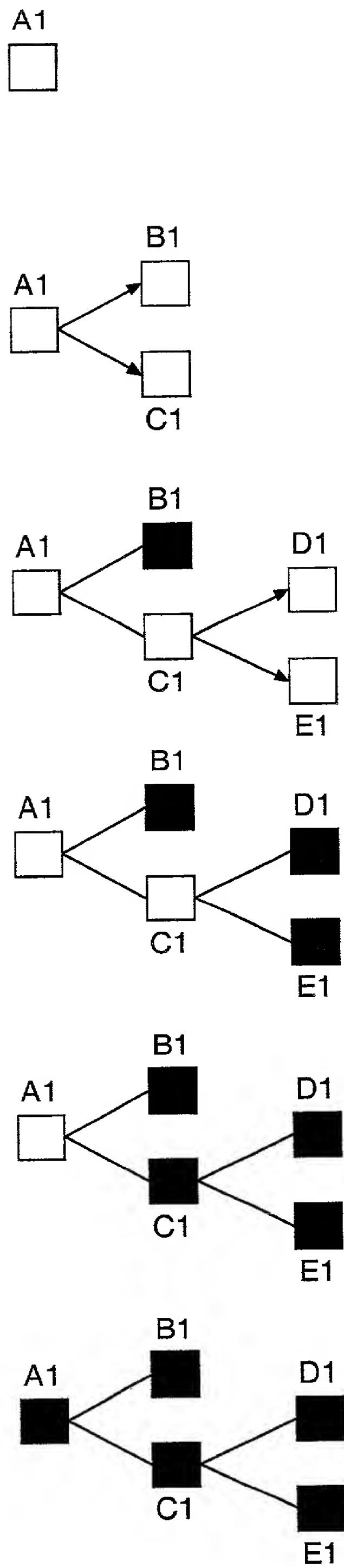
PROCESSING OF SHADOW COMPONENT
DURING COMPONENT LOADING PHASE

Figure 13



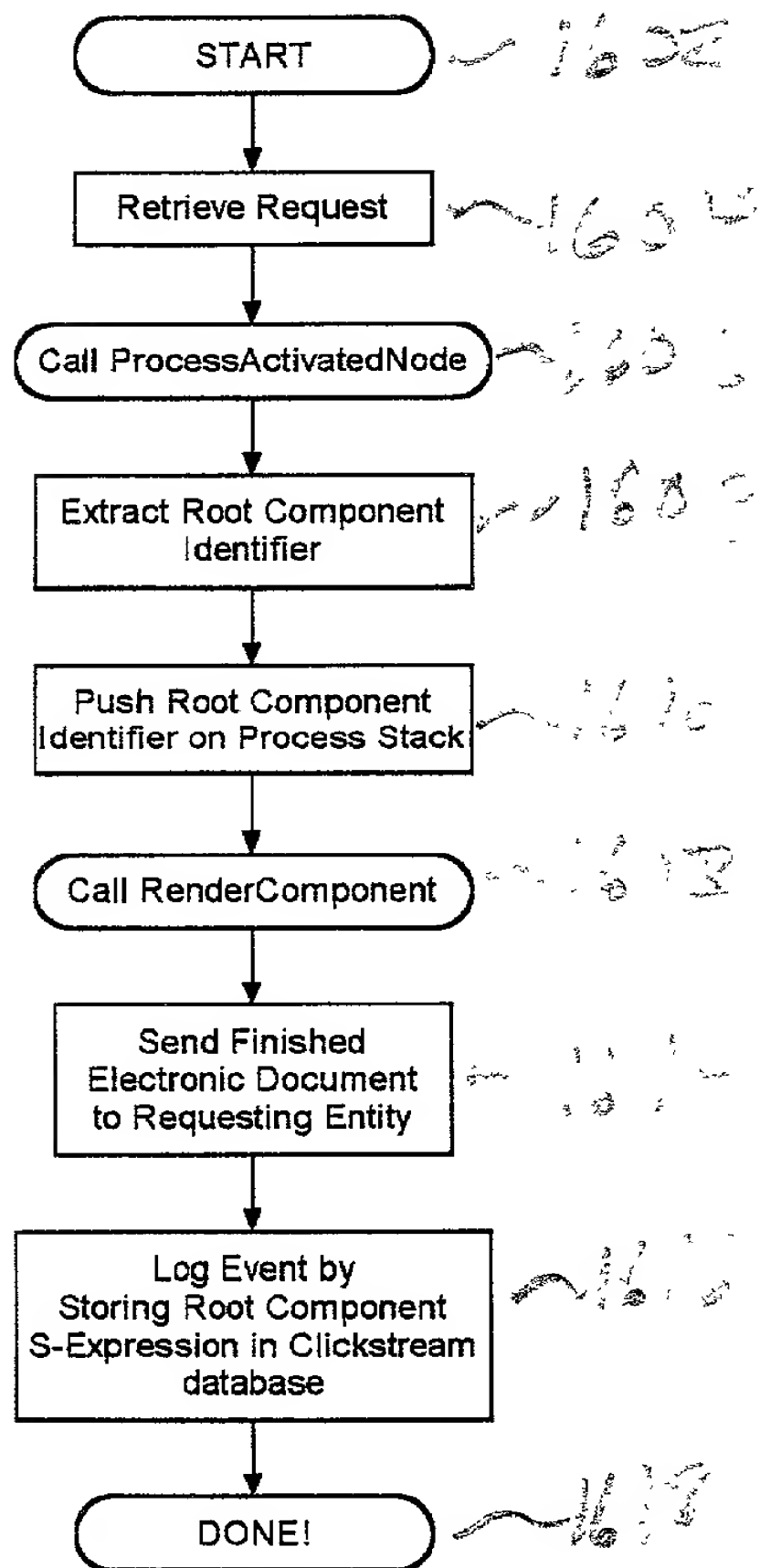
SHADOW COMPONENT SELF-MODIFICATION

Figure 14



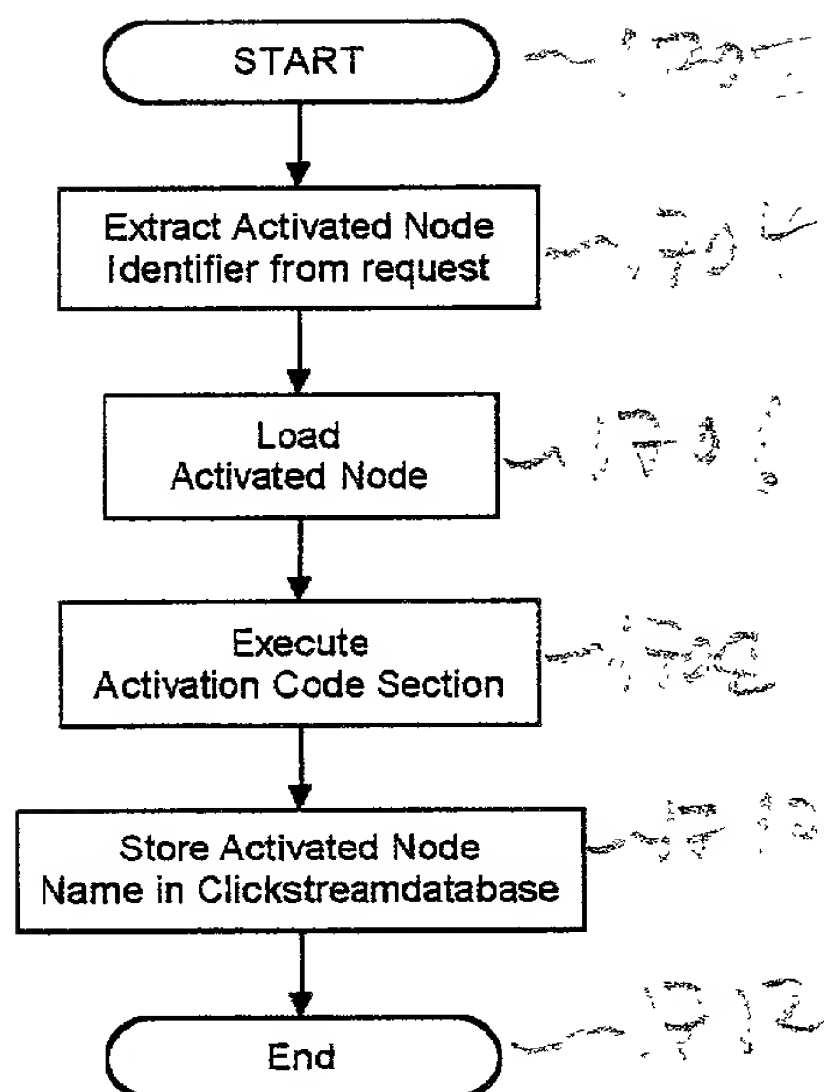
MULTITHREADED COMPONENT LOADING AND PROCESSING

Figure 15



COMPONENT LOADING & PROCESSING USING RECURSION
MAIN ROUTINE

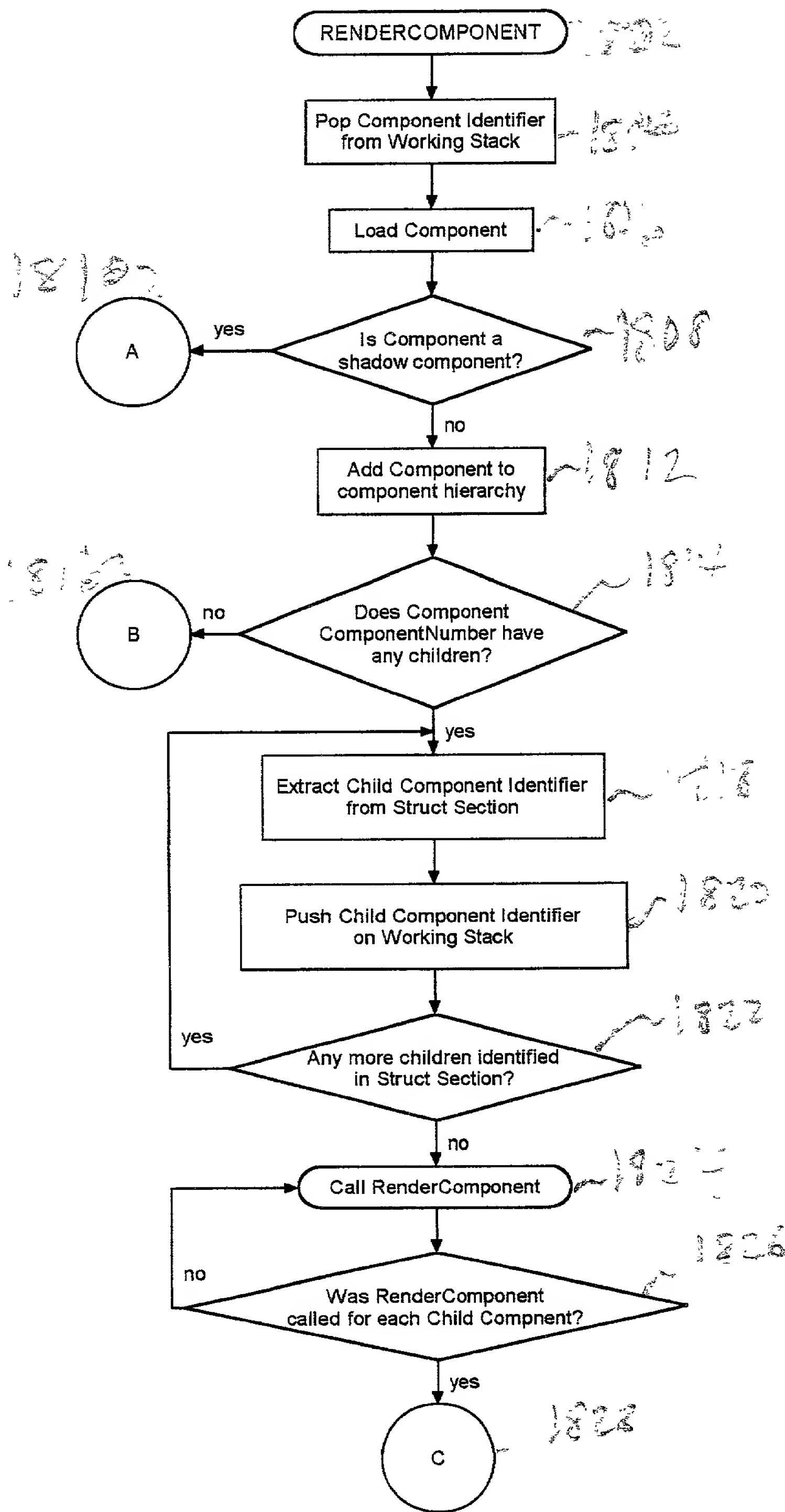
Figure 16



COMPONENT LOADING & PROCESSING USING RECURSION
ROUTINE PROCESSACTIVATEDNODE

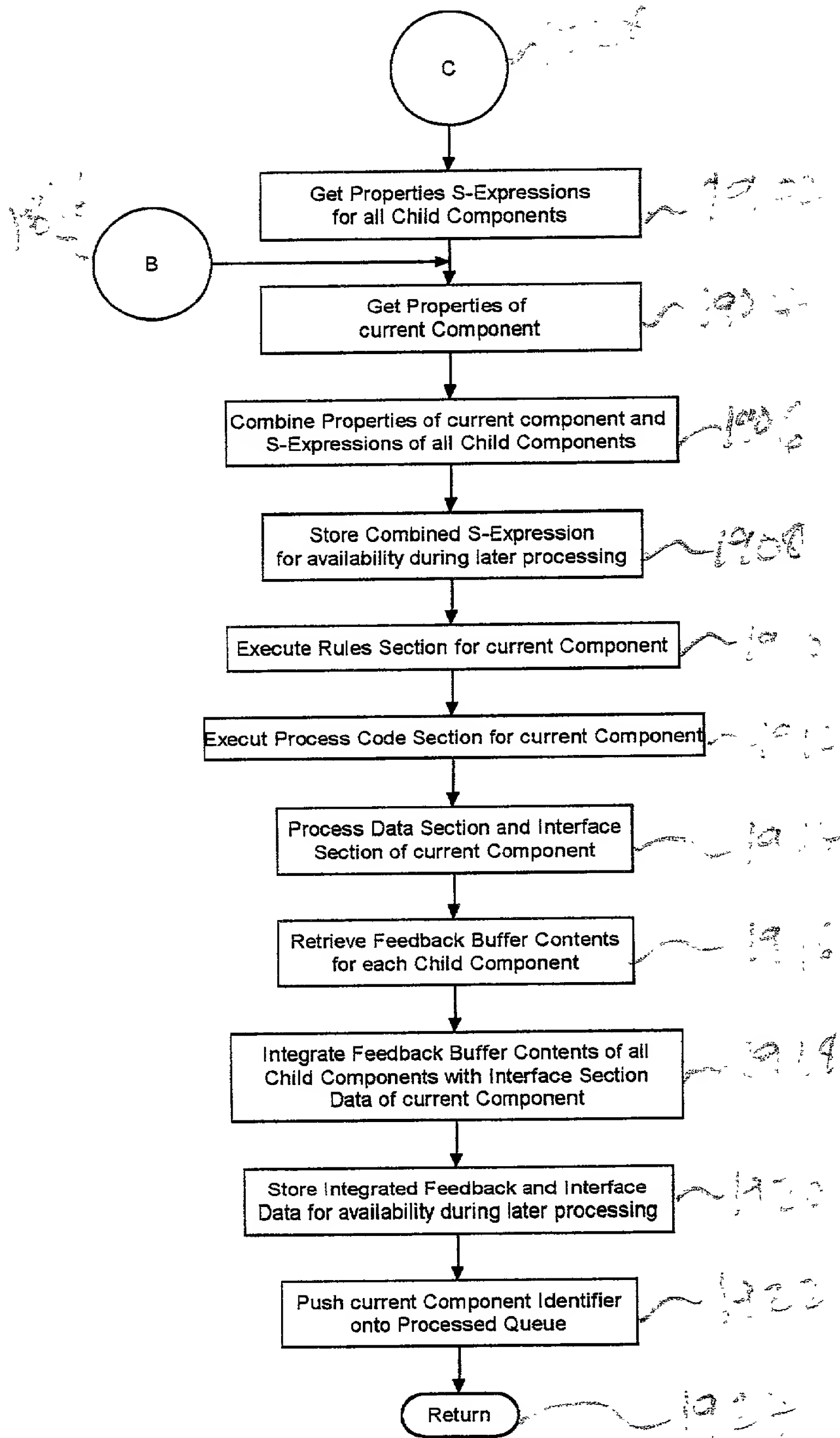
Figure 17

098334 06340



COMPONENT LOADING & PROCESSING USING RECURSION
RENDERCOMPONENT ROUTINE (PART 1/3)

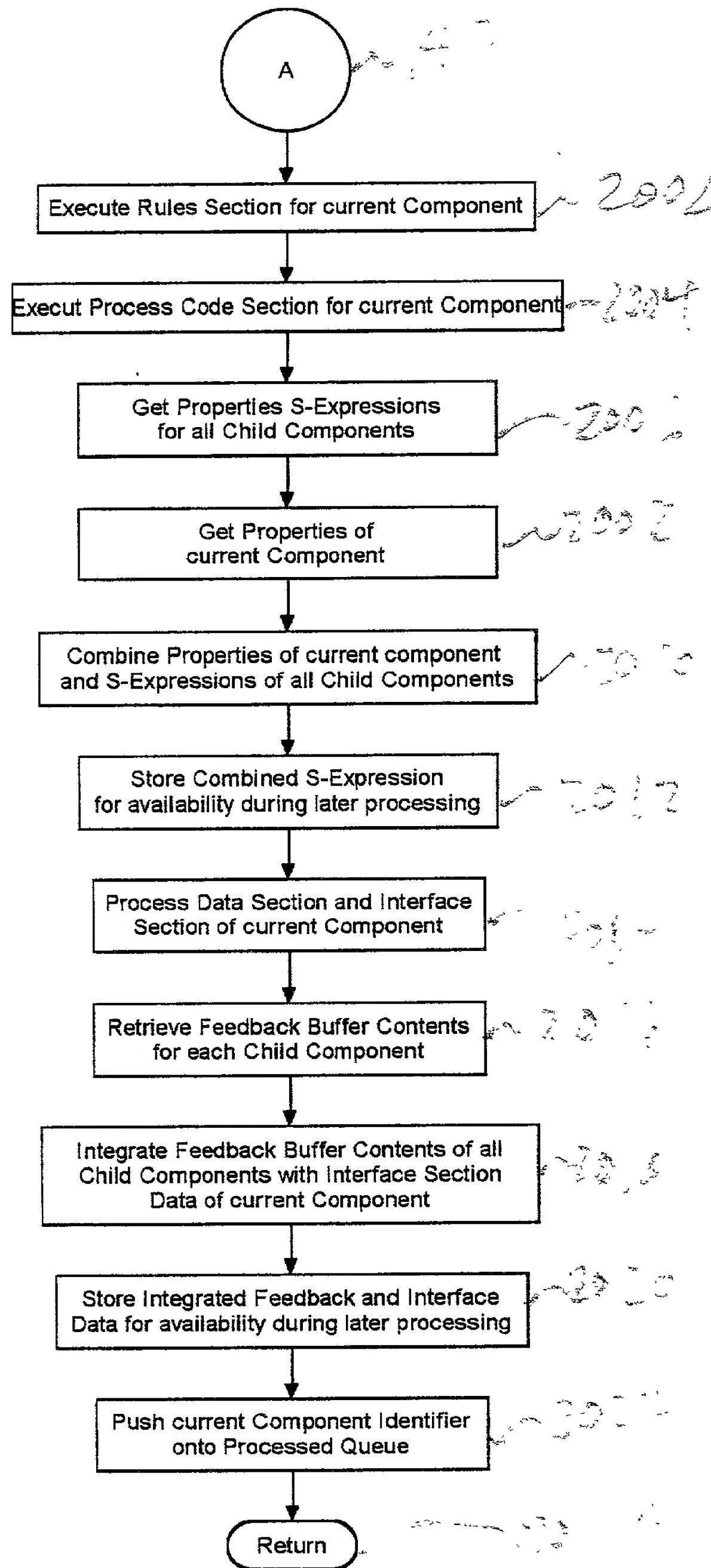
Figure 18

[illegible]

COMPONENT LOADING & PROCESSING USING RECURSION

RENDERCOMPONENT ROUTINE (PART 2/3)

Figure 19



COMPONENT LOADING & PROCESSING USING RECURSION
RENDERCOMPONENT ROUTINE (PART 3/3)

Figure 20

EXAMPLE PROCESSED QUEUE

Figure 21

```
<HTML>
<BODY>
<CENTER><H1>ALOHA'S WEBPAGE</H1></CENTER>
<TABLE>
<TBODY>
<TR>
<TD>
<IMG SRC="aha.html"></IMG>
</TD>
<TD>
Aloha was born ...
</TD>
</TR>
</TBODY>
</TABLE>
</BODY>
</HTML>
```

EXAMPLE FEEDBACK BUFFER

Figure 22

Component

NAME	2302
FAMILY	2304
STRUCT	2306
PROPERTIES	2308
DATA	2310
INTERFACE	2312
PROCESS CODE	2314
ACTIVATION CODE	2316
RULES	2318
VARIABLES	2320

Figure 23

```

<identification> 2402
  <name value="D1"/> 2404
  <type value="normal"/> 2406
  <description value="my picture"/> 2408
</identification> 2410
  
```

EXAMPLE NAME ELEMENT

Figure 24

```

<relatives> 2502
  <relative value="pictures_of_me"/> 2504
  <relative value="home_images"/> 2506
</relatives> 2508
  
```

EXAMPLE FAMILY ELEMENT

Figure 25

```
<struct> 2602
  <reference value="D1"/> 2604
  <reference value="E1"/> 2606
  <reference value="this"/> 2608
</struct> 2610
```

EXAMPLE STRUCT ELEMENT

Figure 26

```
<properties> 2702
  <property name="pizza" value="1.0"/> 2704
  <property name="hot food" value="1.0"/> 2706
</properties> 2708
```

EXAMPLE PROPERTIES ELEMENT

Figure 27


```

<code language="JavaScript">
  <![CDATA[
    // @API
    var fileArr = API.getFileListing("w", "xml");
    var i;
    var newStr = "";
    if (fileArr != null) {
      for (i=0; i<fileArr.length; i++) {
        newStr += fileArr[i] + "<br>";
      }
    }
    newStr += "[DONE]";
    API.setComponentInterface(cID, "text/html", newStr);
  ]>
</code>

```

EXAMPLE PROCESS CODE ELEMENT

Figure 30

```

<activationcode language="JavaScript">
  <![CDATA[The user has selected component D1!]]>
</activationcode>

```

EXAMPLE ACTIVATION CODE ELEMENT

Figure 31

3202
<ruleset>
 <name value="oriental_food_preferences_ruleset"/>
</ruleset> *3204*
3206

EXAMPLE RULES ELEMENT

Figure 32

3302
<variables>
 <variable name="userid" value="bezuwork100">
 <variable name="last_visit_datetime" value="0101232048">
 <variable name="this_visit_datetime" value="0106201613">
</variables> *3304*
3306 *3308*

EXAMPLE VARIABLES ELEMENT

Figure 33

73633 T23860

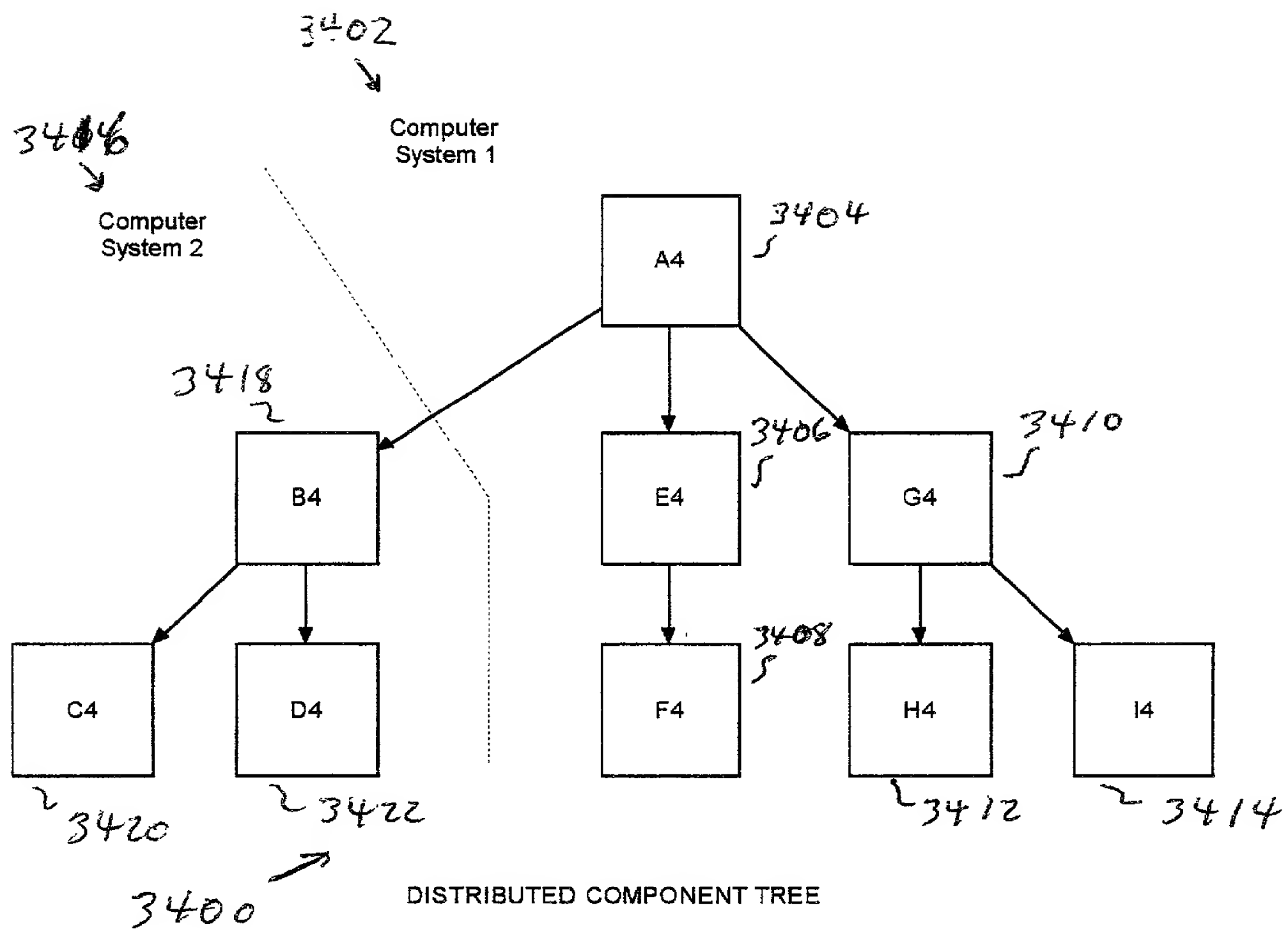
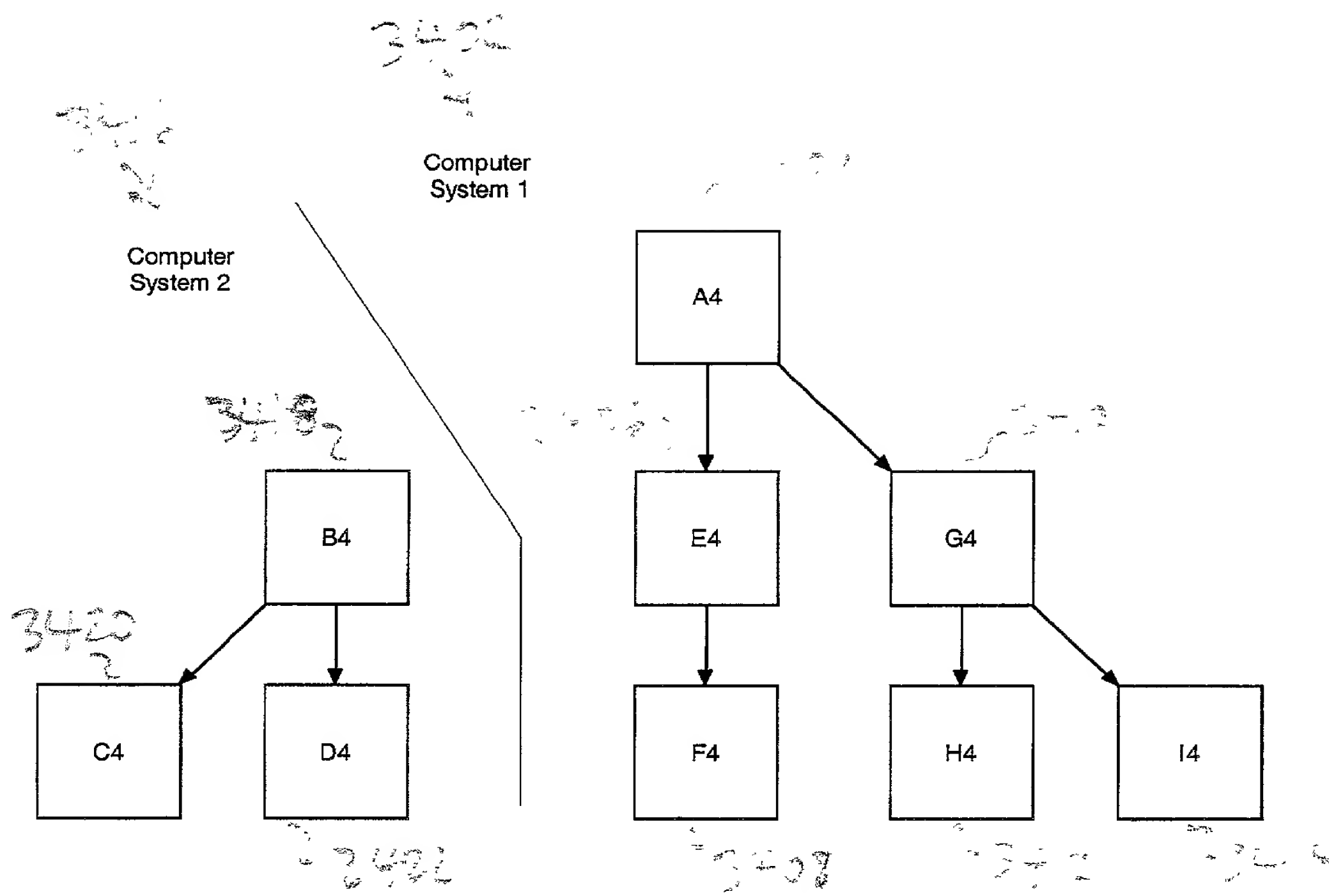


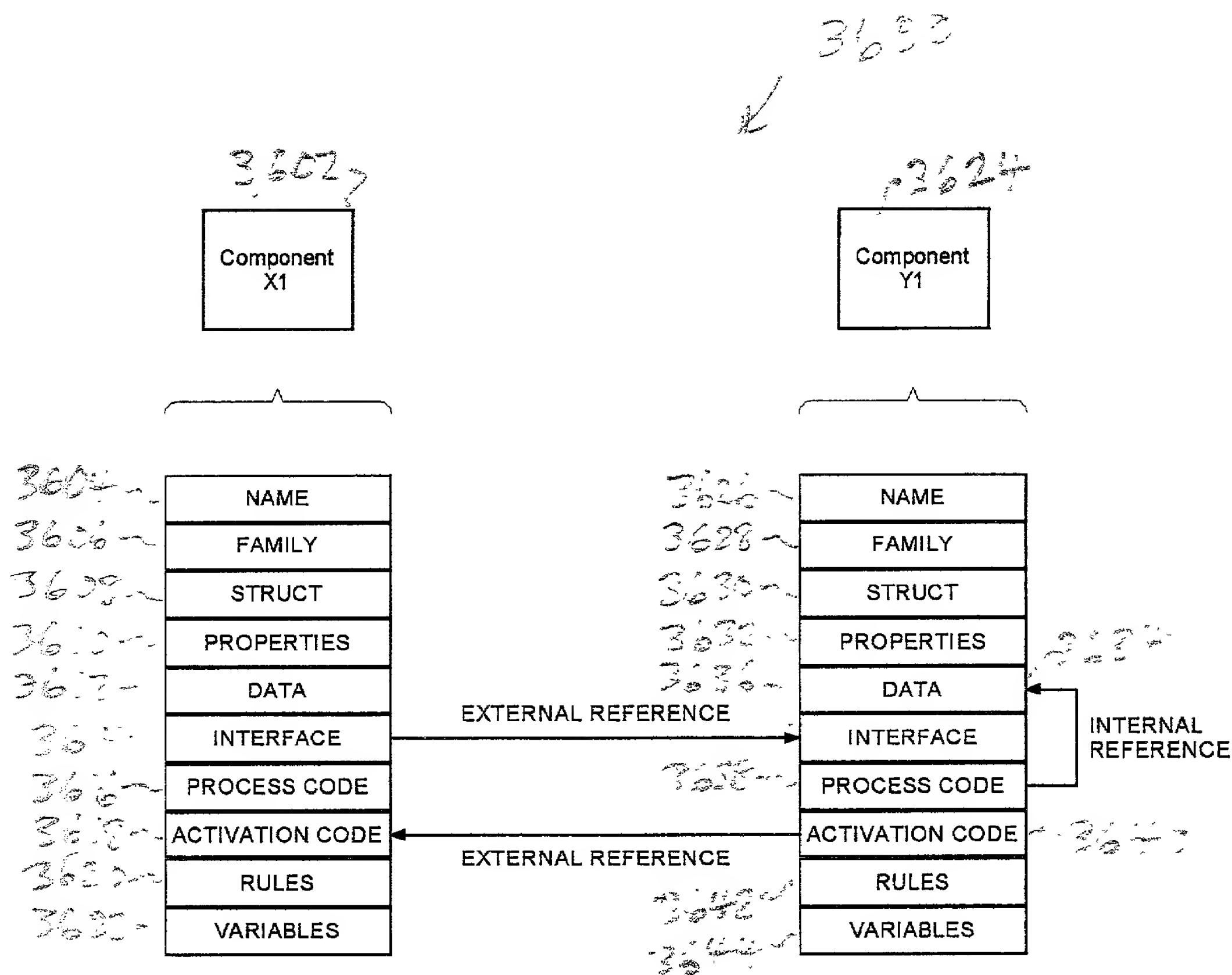
Figure 34.

TOP SECRET



LOADING AND PROCESSING OF DISTRIBUTED COMPONENT TREE

Figure 35



INTERNAL AND EXTERNAL REFERENCING

Figure 36

3700

Yummy Yummy Pizza ✓ 3702

ONLINE DELIVERY ORDER PLACEMENT ✓ 3704

PLEASE SELECT DESIRED ITEM ✓ 3706

PIZZA	OVEN PREPARED DELICACIES (OTHER THAN PIZZA)	HOT & COLD SUBS
SIDE ORDERS	HOMEPAGE CONTACT YUMMY YUMMY	DRINKS
	EVENTS MESSAGE BOARD	

3708 ✓ 3714 ✓ 3716 ✓ 3720 ✓ 3724 ✓ 3718 ✓ 3722 ✓

EXAMPLE INITIAL ONLINE ORDERING PAGE

Figure 37

2500

Yummy Yummy Pizza3807

ONLINE DELIVERY ORDER PLACEMENT3804

DESIGN YOUR OWN PIZZA - PLEASE SELECT SIZE3006

PERSONAL
(8 inch)

3808

MEDIUM
(14 inch)

38103814

LARGE
(18 inch)

3812

GIGANTIC
(24 inch)

3816

ABORT

3818

EXAMPLE SECOND ONLINE ORDERING PAGE

Figure 38

3900

Yummy Yummy Pizza 3902

ONLINE DELIVERY ORDER PLACEMENT 3904

DESIGN YOUR OWN PIZZA - PLEASE SELECT TYPE 3906

THICK CRUST

THIN CRUST

PAN

3908

3910

3912

3914
ABORT

EXAMPLE THIRD ONLINE ORDERING PAGE

Figure 39

4000

Yummy Yummy Pizza 4002

ONLINE DELIVERY ORDER PLACEMENT 4001

DESIGN YOUR OWN PIZZA - PLEASE SELECT TOPPINGS 4006

SAUSAGE 4003	PEPPERONI 4010	HAM 4015	CHICKEN 4021
MUSHROOM 4016	TOMATO 4018	PINEAPPLE 4020	BLACK OLIVES 4022
ONIONS 4024	GREEN PEPPERS 4026	EGGPLANT 4028	EXTRA CHEESE 4030
PEPPER 4032	SALT 4034	ROSEMARY 4036	ABORT 4038
TOPPING SELECTED SO FAR: 4040			DONE! 4042

EXAMPLE FOURTH ONLINE ORDERING PAGE

Figure 40

4100

4102

4104

4106

4108

4110

Yummy Yummy Pizza

ONLINE DELIVERY ORDER PLACEMENT

CURRENT CONTENTS OF YOUR SHOPPING CART:

	PRICE
PIZZA # 1:	15.00
NUMBER: 1	
SIZE: GIGANTIC	
TYPE: THICK CRUST	
TOPPINGS: CHICKEN, PEPPER	2X1.50= 3.00
SIDE #1: BREADSTICKS, NUMBER: 1 ORDER	1.50
DRINKS: COKE, CAN, NUMBER: 1 SIXPACK	6.00
	TAX: 1.80
	TOTAL: 27.30

4108

CONTINUE
ITEM
SELECTION

PROCESS
ORDER

EXAMPLE CONTENTS OF ONLINE SHOPPING CART

Figure 41

4203 (A1 (B1) (C1 (D1) (E1)))
4204 (D1)

CLICKSTREAM DATABASE WITHOUT USER ID OR PROPERTIES

Figure 42

Yummy Yummy Pizza
ONLINE DELIVERY ORDER PLACEMENT

SELECT ITEM, ENTER NUMBER OF UNITS DESIRED, THEN CLICK "ADD TO ORDER"
WHEN DONE, CLICK "PROCESS ORDER" - YOU WILL HAVE AN OPPORTUNITY TO
REVIEW YOUR ORDER FOR ACCURACY BEFORE SUBMISSION.

HOT OVEN PRODUCTS & SUBS

- ☐ STROMBOLI ☐ CHEESE STEAK SUB ☐ ITALIAN SAUSAGE SUB
☐ CALZONE ☐ MEATBALL SUB ☐ ITALIAN COLD CUT SUB
☐ PIZZA

SIZE: ☐ PERSONAL ☐ MEDIUM ☐ LARGE ☐ GIGANTIC

CRUST: ☐ THIN CRUST ☐ THICK CRUST

TOPPING:

- ☐ SAUSAGE ☐ PEPPERONI ☐ HAM ☐ EXTRA CHEESE
☐ MUSHROOM ☐ BLACK OLIVES ☐ TOMATOE ☐ PINEAPPLE
☐ ONIONS ☐ GREEN PEPPERS ☐ EGGPLANT ☐ CHICKEN

HERBS & SPICES: ☐ PEPPER ☐ SALT ☐ ROSEMARY ☐ OREGANO

SIDE ORDERS

- ☐ BREADSTICKS ☐ BUFFALO WINGS ☐ POTATOE CHIPS

DRINKS

- ☐ LITER COKE ☐ SPRITE CAN ☐ ICED TEA, BOTTLE
☐ COKE, CAN ☐ DIET COKE, CAN ☐ WATER, BOTTLED

NUMBER OF SELECTED ITEMS DESIRED:

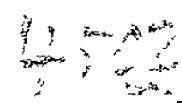
1

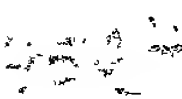
ADD TO ORDER

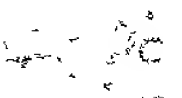
PROCESS ORDER

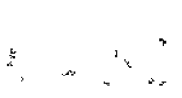
EXEMPLARY WEBPAGE WITH USER FEEDBACK


Figure 44


(webpage)  (user id=bezuwork100 (ZA1 (ZB1) (ZC1) (ZD1) (ZE1) (ZF1) (ZG1, add=1.0) (ZH1, processorder=1.0)))

(response)  (user id=bezuwork100 (ZE1, pizza=1.0, hotfood=.10, gigantic=1.0, chewy=0.9, thickness=0.6, chicken=topping,1.0, pepper=topping,1.0, processorder=1.0))

(webpage)  (user id=bezuwork100 (ZA2 (ZB2) (ZC2) (ZD2) (ZE2) (ZF2) (ZG2, add=1.0) (H2, processorder=1.0)))

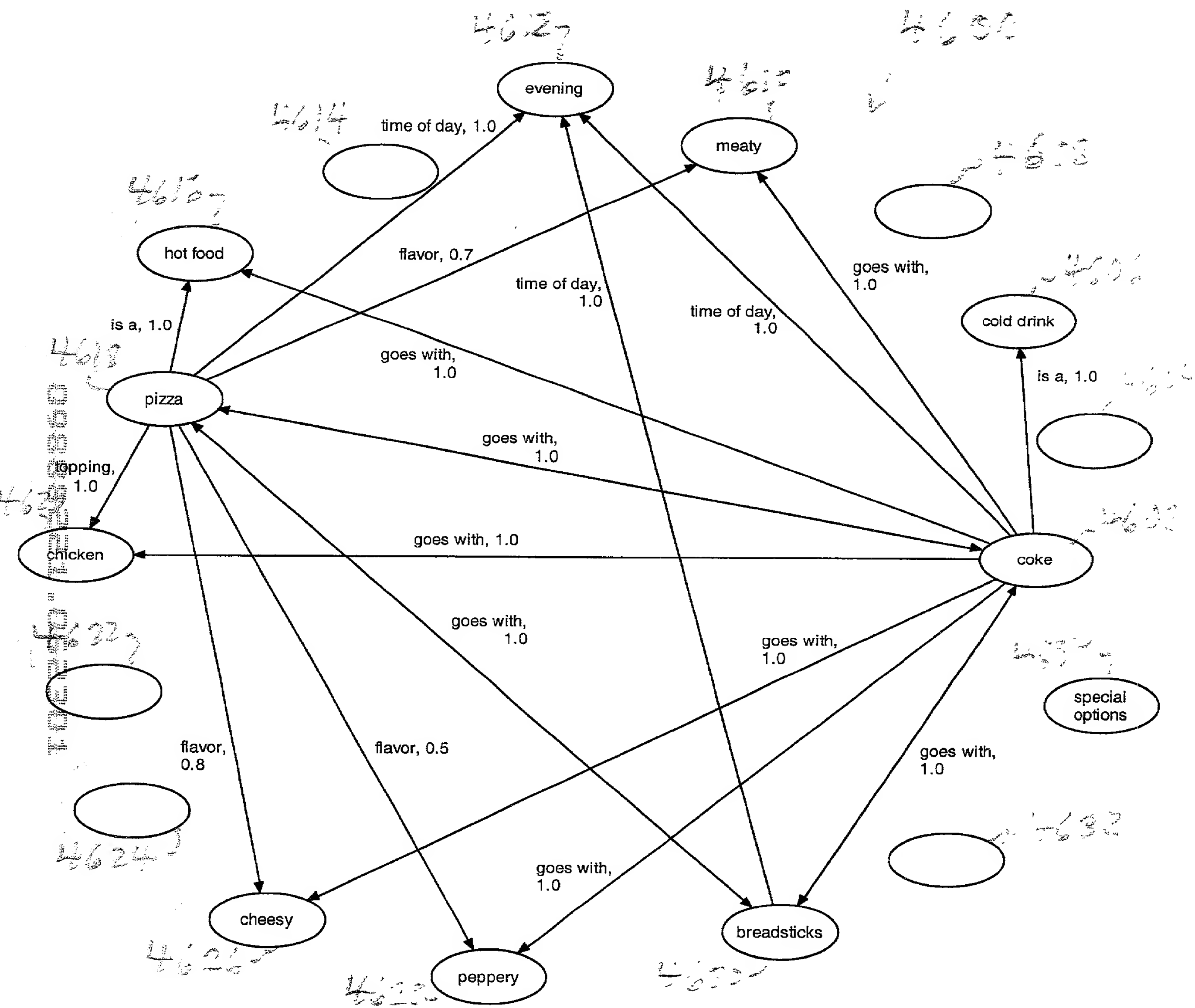
(response)  (user id=bezuwork100 (ZE2 breadsticks=1.0, sideorder=1.0))

(webpage)  (user id=bezuwork100 (ZA3 (ZB3) (ZC3) (ZD3) (ZE3) (ZF3) (ZG3, add=1.0) (ZH3, processorder=1.0)))

(response)  (user id=bezuwork100 (ZE3 coke=1.0, coldrink=is a, 1.0, dataentry value="6"))

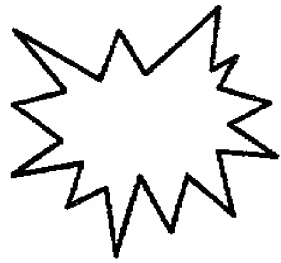
CLICKSTREAM DATABASE WITH PROPERTIES - RESULTING FROM FORM WEBPAGE

Figure 45



SEMANTIC NETWORK REPRESENTATION OF PERSONAL PROFILE

Figure 46



EDITHA'S FILIPINO DINER

1116 South 19th Street
Arlington, VA 22202

For delivery, select menu items below, or go to our delivery page

RESERVATIONS

SPECIALS

FULL MENU

DELIVERY

CATERING

SITE MAP

25

ARROZ CALDONG
(rice & chicken)

21

AFRITADA (chicken & sausage
stew)

34 PAELLA

(chicken & sausage
with rice)

15 GINISANG

MUNGO
(vegetable)

18

PAKIGUNG
ISDA
(fish)

17

ESCABECHE
(fish)

11 ADOBO
(chicken)

12 TORTA
(meat)

15 SINIGANG
NA BABOY
(meat)

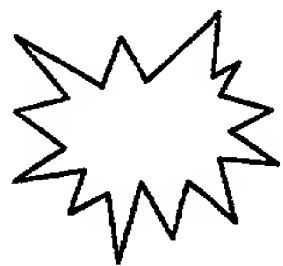
10 PANSIT
(noodles, meat,
& vegetables)

28 TINOLA
MANOK
(chicken)

62 PANDESAL
(bread)

UNCUSTOMIZED WEBPAGE

Figure 47



EDITHA'S FILIPINO DINER

1116 South 19th Street
Arlington, VA 22202

For delivery, select menu items below, or go to our delivery page

DELIVERY

FULL MENU

SPECIALS

CATERING

HISTORY &
LANGUAGE

SITE MAP

42 PAGKAING
BILOG
(pizza Filipino
style)

21 AFRITADA (chicken & sausage
stew)

34 PAELLA

(chicken & sausage
with rice)

101 MANGO
DRINK

103 COCONUT
DRINK

62 PANDESAL
(bread)

11 ADOBO
(chicken)

12 TORTA
(meat)

15 SINIGANG
NA BABOY
(meat)

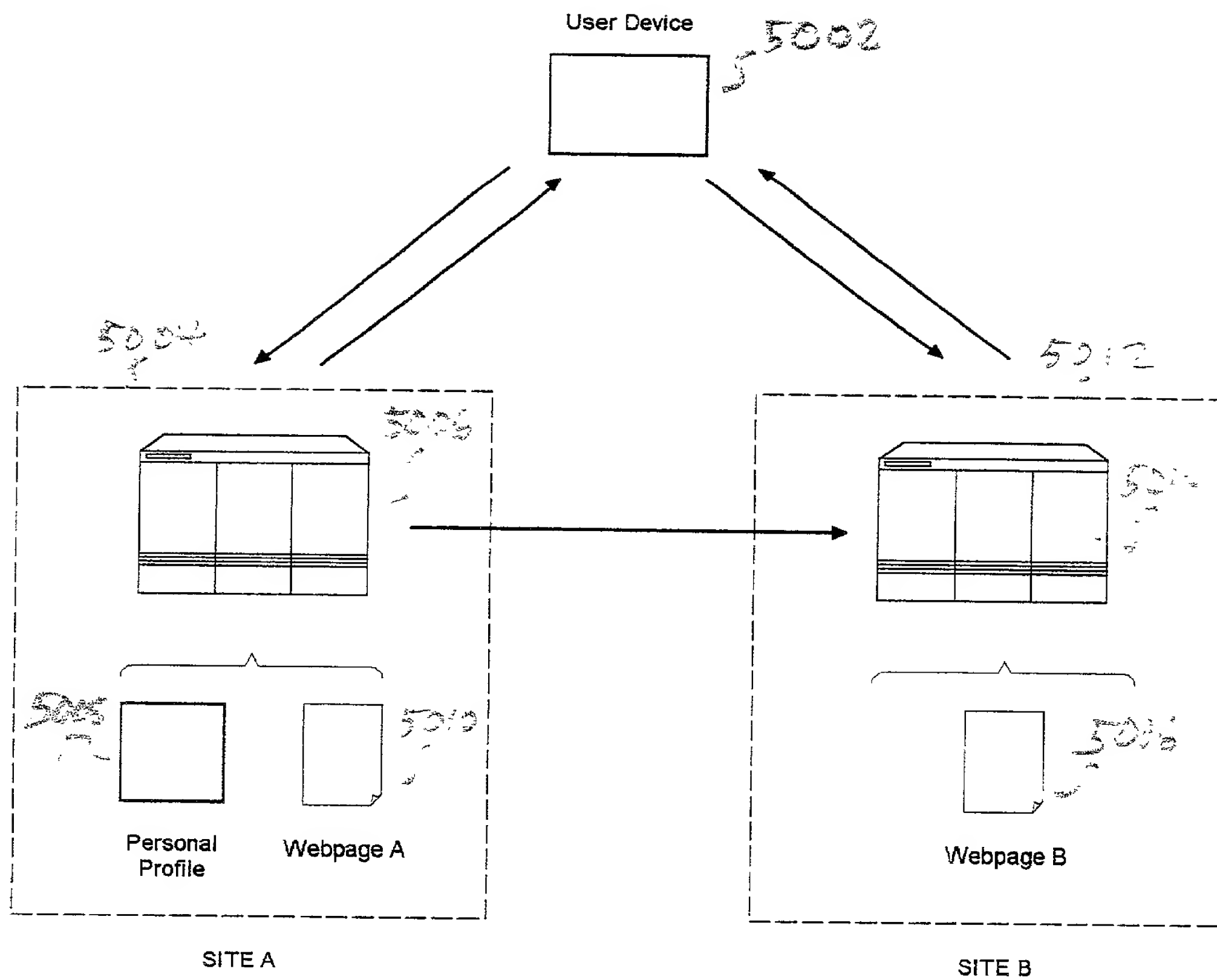
10 PANSIT
(noodles, meat,
& vegetables)

28 TINOLA
MANOK
(chicken)

17 ESCABECHE
(fish)

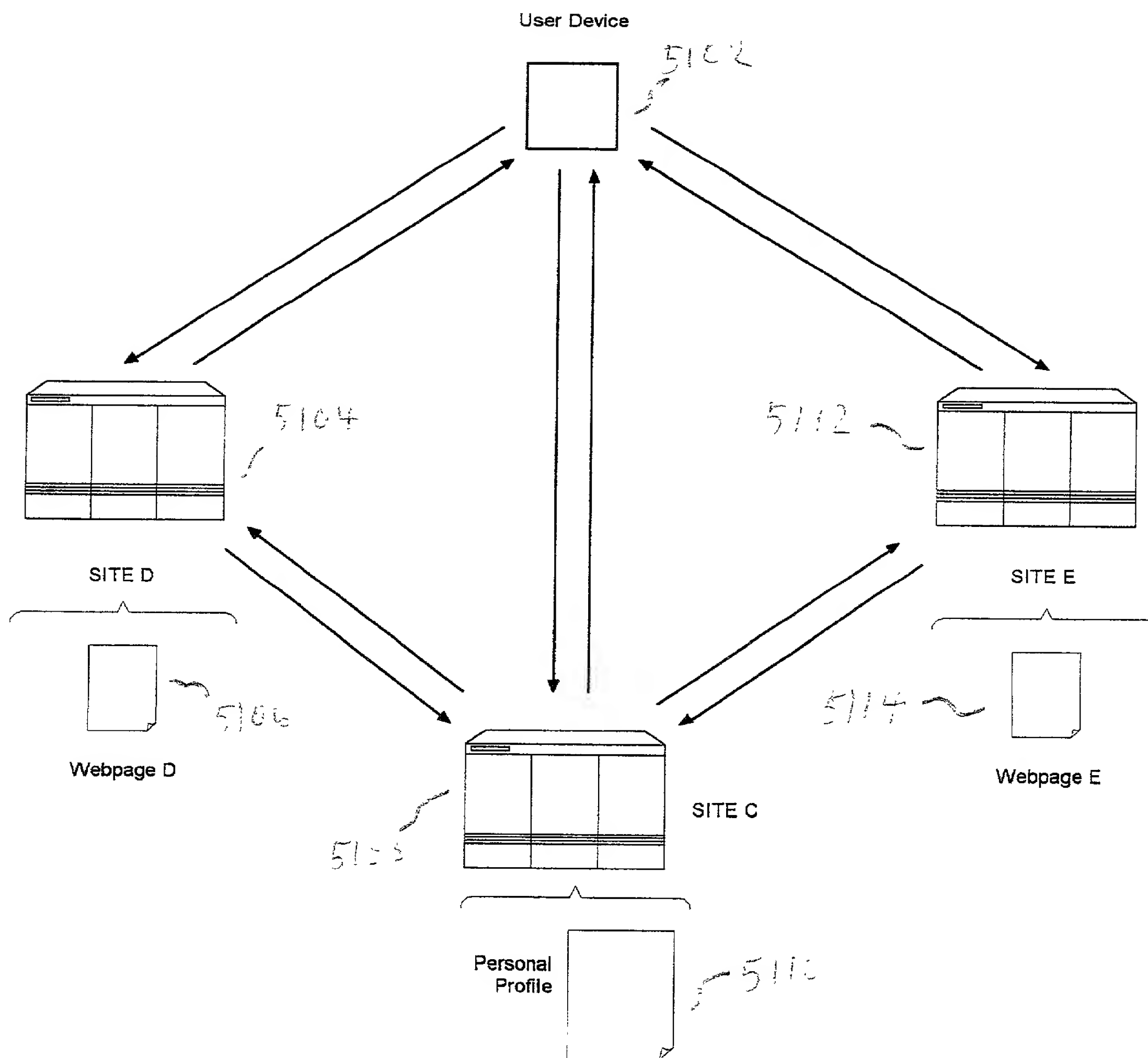
WEBPAGE CUSTOMIZED BY PERSONALIZATION

Figure 48



Decentralized Distribution of Personal Profiles

Figure 50

[illegible]

Centralized Distribution of Personal Profiles

Figure 51

Intellectual Property News Online

SITEMAP

PATENTS

COPYRIGHT

TRADEMARK

TRADE
SECRET

MESSAGE
BOARDS

Youngster's Invention Worth Millions

Young inventor, Aloha invented new technique of surveillance while playing
Bird Diaper - US Patent 5,934,226 seen as boon to bird lovers everywhere

A bird diaper for an uncaged pet bird to wear, featuring an enclosed pouch

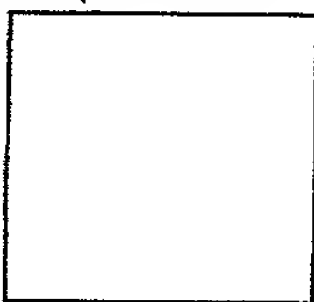
Open Source vs. Microsoft - Who Should Win?

Linus Torvalds, originator of the Linux craze, was recently asked what he had

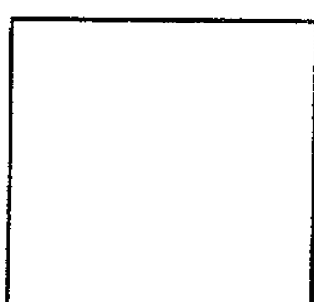
Anit-Cybesquatting law used in test case

The anti-cybersquatting act, signed into law on November 29, 1999, forms the

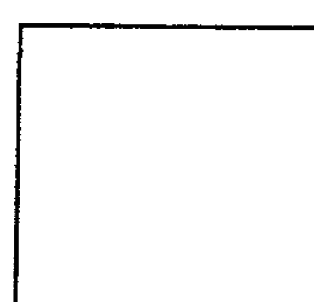
People Profiles



Aloha - Child
Inventor who's
Invention
Interests the
US Military



Abraham
Lincoln -
only US
President to
hold a patent



UNPERSONALIZED WEBPAGE

Figure 52

Intellectual Property News Online

PATENTS

MESSAGE
BOARDS

SITEMAP

TRADEMARK

COPYRIGHT

TRADE
SECRET

Youngster's Invention Worth Millions

Young inventor, Aloha invented new technique of surveillance while playing
Bird Diaper - US Patent 5,934,226 seen as boon to bird lovers everywhere

A bird diaper for an uncaged pet bird to wear, featuring an enclosed pouch

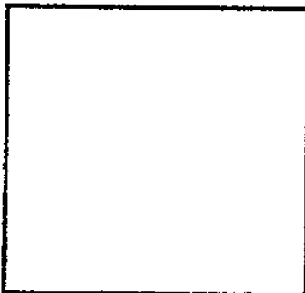
Open Source vs. Microsoft - Who Should Win?

Linus Torvalds, originator of the Linux craze, was recently asked what he had

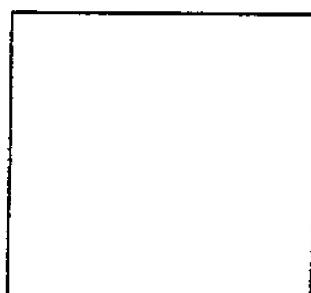
Anit-Cybesquatting law used in test case

The anti-cybersquatting act, signed into law on November 29, 1999, forms the

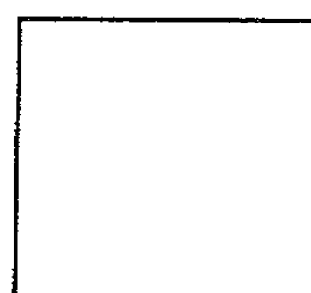
People Profiles



Aloha - Child
Inventor who's
Invention
Interests the
US Military



Abraham
Lincoln -
only US
President to
hold a patent



WEBPAGE WITH REORDERED COMPONENTS

Figure 53

Intellectual Property News Online

SITEMAP

PATENTS

COPYRIGHT

TRADEMARK

TRADE
SECRET

MESSAGE
BOARDS

Youngster's Invention Worth Millions

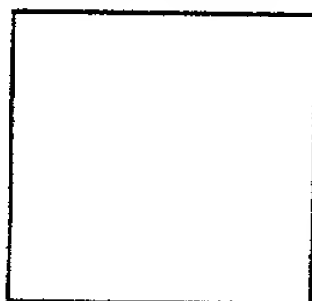
Young inventor, Aloha invented new technique of surveillance while playing
Bird Diaper - US Patent 5,934,226 seen as boon to bird lovers everywhere

A bird diaper for an uncaged pet bird to wear, featuring an enclosed pouch
Festo Decision Causes Patent Practitioners To Reexamine Their Technique

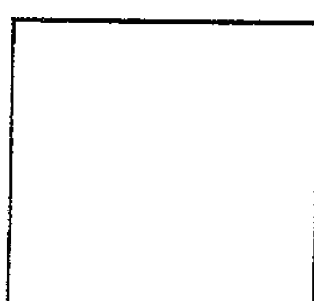
The US Court of Appeals for the Federal Circuit decision in Festo Corp. v. Sho
State Street Bank Decision Seen to open doors for Business Method Patent

In State Street, the US Court of Appeals rejected the business methods excep

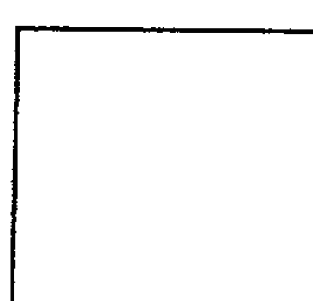
People Profiles



Aloha - Child
Inventor who's
Invention
Interests the
US Military



Abraham
Lincoln -
only US
President to
hold a patent



WEBPAGE SHOWING SELECTIVE FILTERING

Figure 54

098824.052304

5500

Intellectual Property News Online

5202

5234

5204

5207

5207

5207

5207

SITEMAP

PATENTS

COPYRIGHT

TRADEMARK

TRADE
SECRET

MESSAGE
BOARDS

Youngster's Invention Worth Millions

Young inventor, Aloha invented new technique of surveillance while playing
Bird Diaper - US Patent 5,934,226 seen as boon to bird lovers everywhere

A bird diaper for an uncaged pet bird to wear, featuring an enclosed pouch
Open Source vs. Microsoft - Who Should Win?

Linus Torvalds, originator of the Linux craze, was recently asked what he had
Anit-Cybesquatting law used in test case

The anti-cybersquatting act, signed into law on November 29, 1999, forms the
Festo Decision Causes Patent Practitioners To Reexamine Their Technique

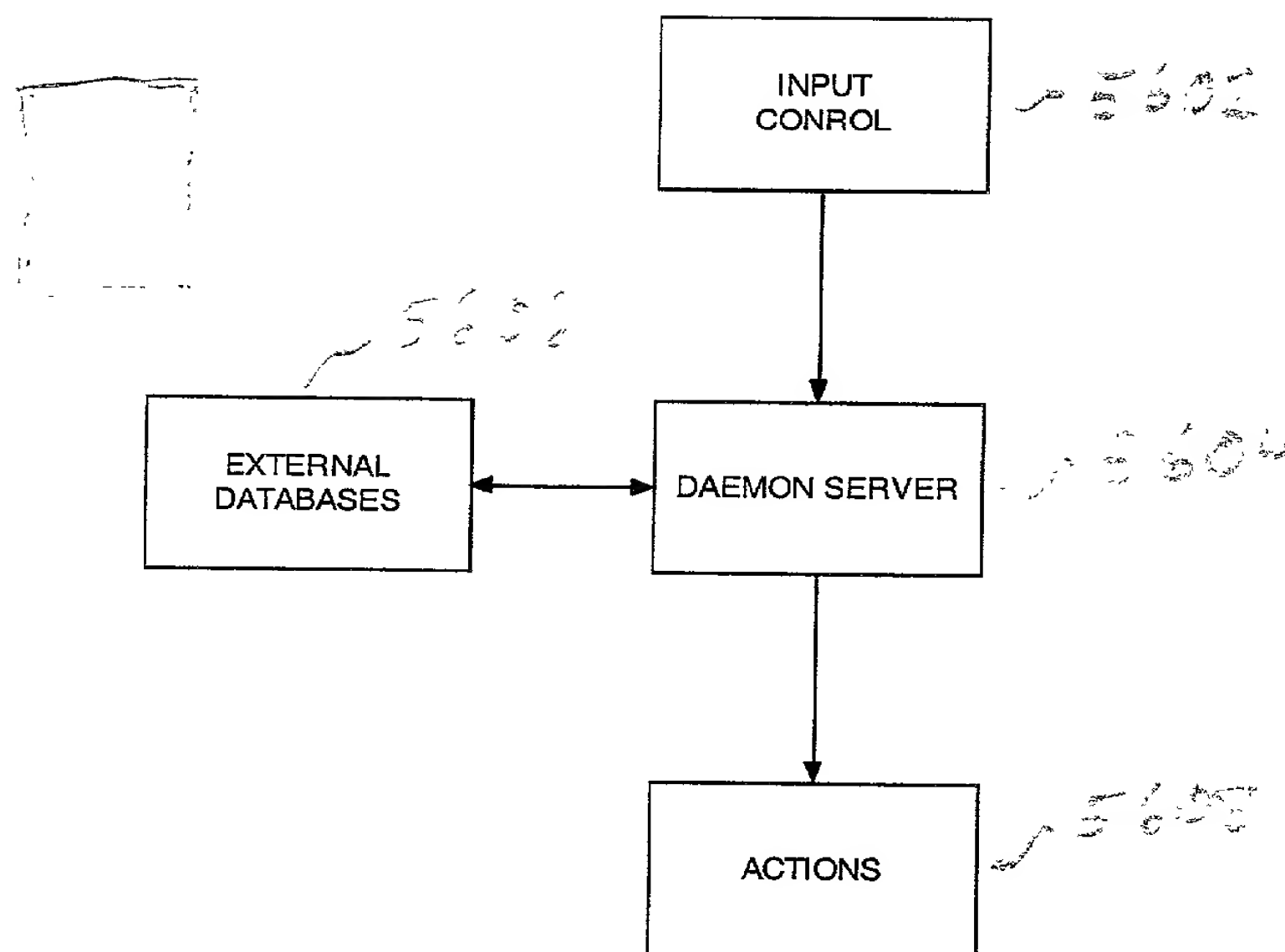
The US Court of Appeals for the Federal Circuit decision in Festo Corp. v. Sho
State Street Bank Decision Seen to open doors for Business Method Patent

In State Street, the US Court of Appeals rejected the business methods excep
In re Beauregard decision paves the way for Program Product claims

The US Patent & Trademark Office changed it's position, deciding not to pursu

WEBPAGE AFTER COMPONENT REMOVAL

Figure 55

[illegible]

DAEMON SERVER

Figure 56

5700

Yummy Yummy Pizza

5702

5704

5706

ONLINE DELIVERY ORDER PLACEMENT

PLEASE SELECT DESIRED ITEM

PIZZA	OVEN PREPARED DELICACIES (OTHER THAN PIZZA)	SPECIAL 20% OFF ANY GIGANTIC PIZZA
SIDE ORDERS	HOMEPAGE	CONTACT YUMMY YUMMY
	EVENTS	MESSAGE BOARD
		DRINKS

5710

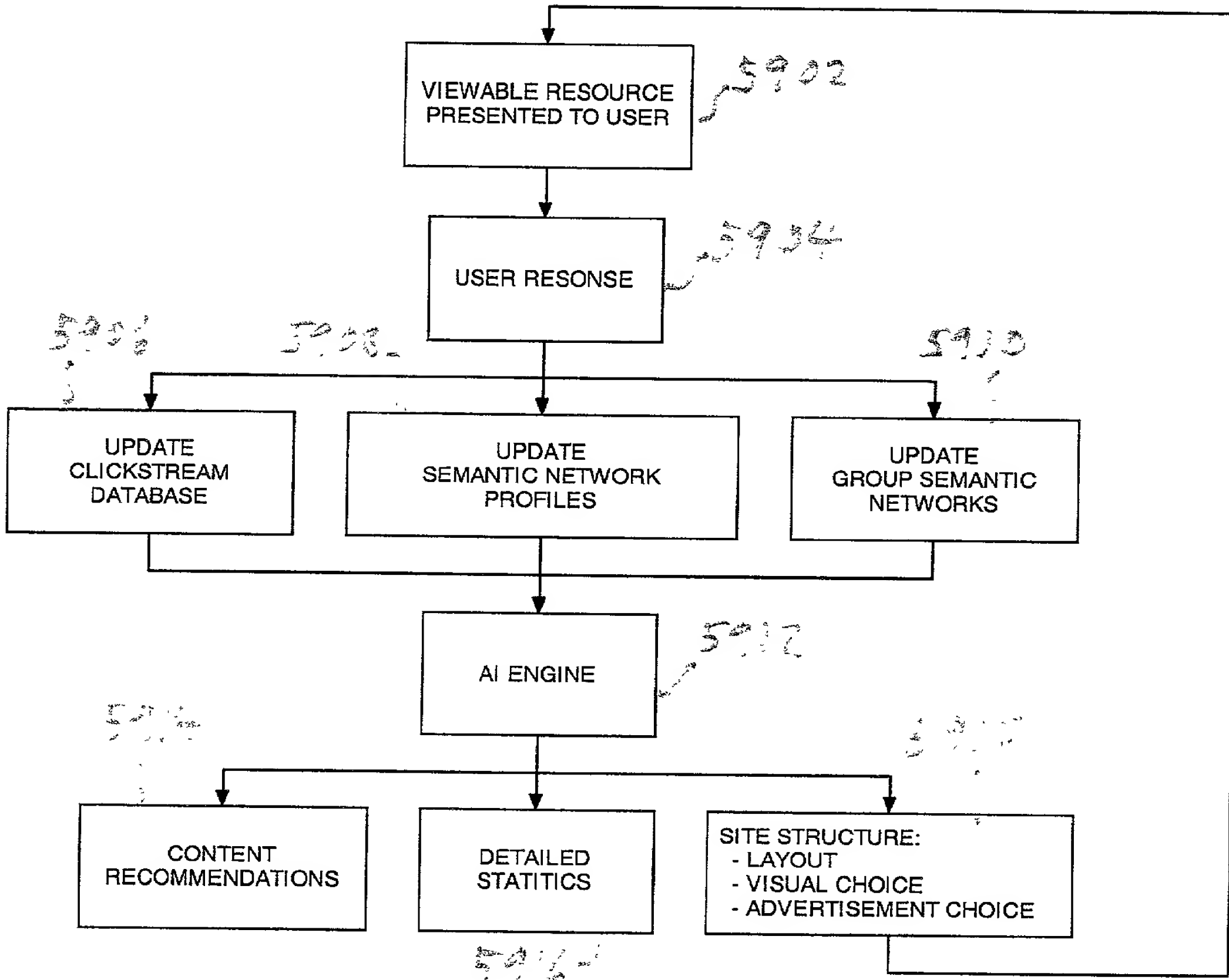
5712

5714

5716

EXAMPLE INITIAL ONLINE ORDERING PAGE
WITH EXAMPLE OF EXTRA-SESSION PERSONALIZATION

Figure 57



OPERATION OF THE ARTIFICIAL INTELLIGENCE ENGINE

Figure 58

The diagram illustrates a network topology for content delivery. At the top center is a **RESOURCE SERVER**, represented by a server rack icon. Two lines extend from the Resource Server to two separate paths of intermediate servers, each represented by a small square box. The path on the left consists of three boxes connected in series, leading to a **USER DEVICE** (represented by a rectangle) at the bottom left. The path on the right also consists of three boxes connected in series, leading to a **LOCAL CACHE SERVER** (represented by a larger rectangle) and then to another **USER DEVICE** at the bottom right. The labels **RESOURCE SERVER**, **LOCAL CACHE SERVER**, and **USER DEVICE** are placed near their respective components. There are several handwritten-style scribbles or marks scattered around the diagram, possibly indicating specific nodes or connections.

Figure 59

```

<?xml version="1.0"?>

<componentDefinition>
  <identification>
    <name value="componentSprint"/>
    <type value="normal"/>
  </identification>
  <relatives>
    <relative value="componentSprint"/>
  </relatives>
  <struct>
    <reference value="this"/>
  </struct>
  <properties>
  </properties>
  <data type="none">
    <![CDATA[]]>
  </data>
  <interface type="text/html">
    <![CDATA[]]>
  </interface>
  <code language="WebL">
    <![CDATA[
      import Java;
      // @CCID
      // Define Java API interface
      // Specify the particular Method that contains the routines for
the API calls
      var jclass = Java_New("Future4");
      var API = jclass.getOCPDS(ccID);
      // Begin
      var carrierName = "Sprint PCS";
      var baseUrl =
"http://www.messaging.sprintpcs.com/sms/check_message_syntax.html?callbacknum=
7777777777";
      var messageHeader = "&message=";
      var defaultMessage = "This+is+a+sample+message";
      var message = API.gOQV("message");
      var mobileHeader = "&mobilenumber=";
      var defaultMobileNum = "7038631132";
      var mobileNum = API.gOQV("mobileNumber");
      if message == "" or message == nil then
        message = defaultMessage;
      end;
      if mobileNum == "" or mobileNum == nil then
        mobileNum = defaultMobileNum;
      end;
      var P = GetURL(baseUrl + messageHeader + message + mobileHeader +
mobileNum);
      var myOutput = carrierName + " message sent.";
      API.sCI(cID, "text/html", myOutput);
    ]]>
  </code>
  <activationCode language="JavaScript">
    <![CDATA[]]>
  </activationCode>
</componentDefinition>

```

exemplary component definition - ComponentSprint

Figure 60

```
<?xml version="1.0"?>
<componentDefinition>
  <identification>
    <name value="ComponentRU"/>
    <type value="preex"/>
  </identification>
  <ruleset>
    <name value="products"/>
  </ruleset>
  <relatives>
  </relatives>
  <struct>
    <reference value="this"/>
  </struct>
  <properties>
  </properties>
  <data type="xml">
    <![CDATA[]]>
  </data>
  <interface type="text/html">
    <![CDATA[]]>
  </interface>
  <code language="JavaScript">
    <![CDATA[]]>
  </code>
  <activationCode language="JavaScript">
    <![CDATA[]]>
  </activationCode>
</componentDefinition>
```

exemplary component definition - ComponentRU
Figure 61

```

<?xml version="1.0"?>

<componentDefinition>
  <identification>
    <name value="componentPR"/>
    <type value="normal"/>
  </identification>
  <relatives>
    <relative value="componentPR"/>
  </relatives>
  <struct>
    <reference value="this"/>
  </struct>
  <properties>
  </properties>
  <data type="none">
    <![CDATA[]]>
  </data>
  <bogus>sad;lkjf;saldkjf;dslkfj;alsdkjfa;lskdjfa;lksj</bogus>
  <interface type="text/html">
    <![CDATA[]]>
  </interface>
  <code language="WebL">
    <![CDATA[
      import Java;
      //try
      // @CCID
      // Define Java API interface
      // Specify the particular Method that contains the routines
      for the API calls
      var jclass = Java_New("Future4");
      var API = jclass.getOCPDS(ccID);
      //var P = GetURL("http://www.prnewswire.com/home.shtml");
      var P = GetURL("http://199.230.26.69/home.shtml");
      var table = Elem(P, "table");
      var QA = table[3];
      var QB = Elem(QA, "tr");
      var myOutput = "<IMG ALIGN=Middle
SRC=\"http://www.prnewswire.com/images/logoanim3.gif\" BORDER=\"0\" ALT=\"PR
Newswire\" VSPACE=7>" +
        "<br><table>" +
        Markup(QB[0]) + Markup(QB[1]) + Markup(QB[2]) +
Markup(QB[3]) +
        "</table>";
      API.sCI(cID, "text/html", myOutput);
    //catch E
    //      on true do
    //          "Data timeout / not available."
    //end
    ]]>
  </code>
  <activationCode language="JavaScript">
    <![CDATA[]]>
  </activationCode>
</componentDefinition>

```

exemplary component definition - ComponentPR

Figure 62


```

<?xml version="1.0"?>

<componentDefinition>
  <identification>
    <name value="componentFileListXML"/>
    <type value="normal"/>
  </identification>
  <relatives>
  </relatives>
  <struct>
    <reference value="this"/>
  </struct>
  <properties>
  </properties>
  <data type="none">
    <![CDATA[]]>
  </data>
  <interface type="text/xml">
    <![CDATA[
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html"/>
  <xsl:template match="/">
    <table>
      <xsl:for-each select="/filelist/file">
        <xsl:sort select="fullfilename"/>
        <!--sorted within grouping-->
        <!--member distinctions-->
        <tr>
          <td>
            <font color="#000000" size="1" face="Arial, Helvetica">
              <xsl:element name="a">
                <xsl:attribute name="href">
                  <xsl:text>/servlet/Future4?cid=</xsl:text>
                    <xsl:value-of select="filename"/>
                </xsl:attribute>
                <xsl:value-of select="filename"/>
              </xsl:element>
            </font>
          <br/>
        </td>
        </tr>
      </xsl:for-each>
    </table>
  </xsl:template>
</xsl:stylesheet>
]]>
  </interface>
  <code language="JavaScript">
    <![CDATA[
// @API
var xmlString = API.getFileListingXML("", "xml");
API.setComponentData(cID, "text/xml", xmlString);
]]>
  </code>
  <activationCode language="JavaScript">
    <![CDATA[]]>
  </activationCode>
</componentDefinition>

```

exemplary component definition - ComponentFileListXML

Figure 64